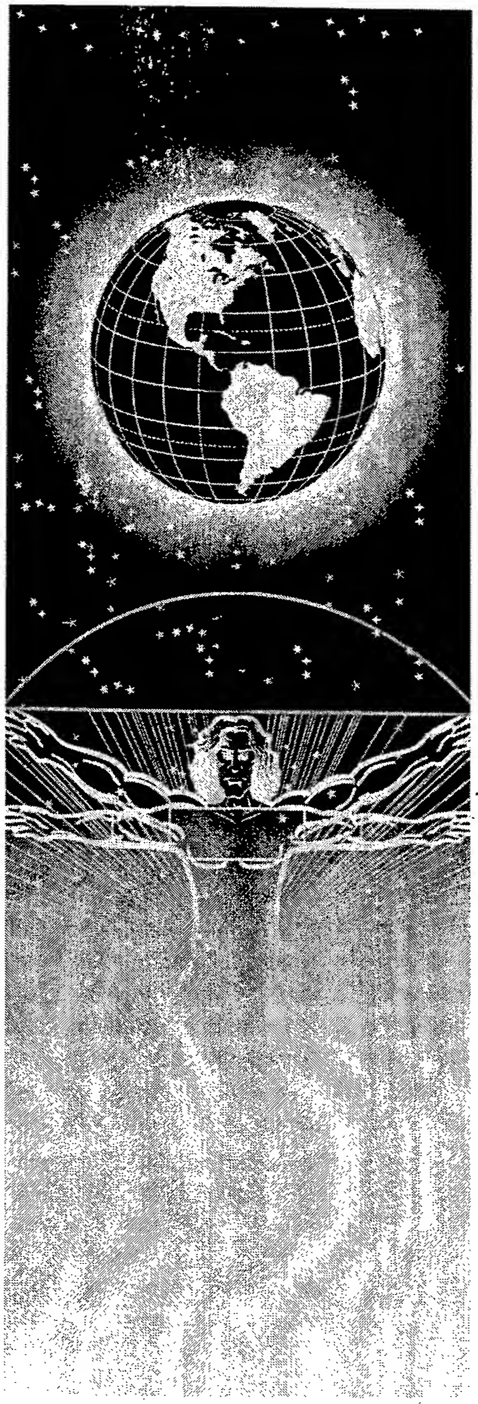
AFRL-HE-WP-TR-2002-0188



UNITED STATES AIR FORCE RESEARCH LABORATORY

ADVANCED INTERFACES AND TESTBED FOR SPACE OPERATOR CONSOLES SBIR PHASE II, FINAL REPORT

Thomas J. Starkey
Thomas W. Marlow
Robert J. Remington

MONTEREY TECHNOLOGIES, INC.
24600 SILVER CLOUD COURT, SUITE 103
MONTEREY CA 93940-6536

John D. Ianni

HUMAN EFFECTIVENESS DIRECTORATE
CREW SYSTEM INTERFACE DIVISION
WRIGHT-PATTERSON AFB OH 45433-7022

APRIL 2002

FINAL REPORT FOR THE PERIOD 13 APRIL 2000 TO 29 AUGUST 2002

Approved for public release; distribution is unlimited

Human Effectiveness Directorate
Crew System Interface Division
2255 H Street
Wright-Patterson AFB OH 45433-7022

NOTICES

DISCLAIMER

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2002-0188

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

MARIS M. VIKMANIS
Chief, Crew System Interface Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
April 2002

2. REPORT TYPE
Final Report

3. DATES COVERED *(From - To)*
13 Apr 2000-29 Aug 2002

4. TITLE AND SUBTITLE
Advanced Interfaces and Testbed For Space Operator Consoles
SBIR Phase II, FINAL REPORT

5a. CONTRACT NUMBER
F33615-00-C-6006

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
65502F

5d. PROJECT NUMBER
3005

5e. TASK NUMBER
HC

5f. WORK UNIT NUMBER
03

6. AUTHOR(S)
*Sharkey, Thomas J., Marlow, Thomas W., Remington, Robert J., and **Ianni, John D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
*Monterey Technologies, Inc.
24600 Silver Cloud Court
Suite 103
Monterey, CA 93940-6536

8. PERFORMING ORGANIZATION REPORT NUMBER
MTI 02.200304

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
**Air Force Research Laboratory
Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB OH 45433-7022

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL-HE-WP-TR-2002-0188

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report summarizes the development of an improved Human-System Interface (HSI) for space operators, and the development of a testbed for evaluating this and other new HSI concepts. This work was performed under a Small Business Innovation Research (SBIR) contract to Monterey Technologies, Inc. A task analysis was performed to identify bottlenecks in the current HSI. It was found that controllers were limited by the number of manual and visual actions required, by the organization and presentation of the data from the satellite, and limited availability of tools that aid the controller. The design of the HSI focused on elimination of these bottlenecks. The HSI developed to address these issues is multi-modal. It features voice synthesis and recognition, touchscreen capabilities, as well as a conventional keyboard and mouse interface. This HSI was implemented in a testbed that allows collection of controller performance data during a simulated support of a DSCS type satellite. The testbed architecture was designed to facilitate integration and evaluation of new, emerging HSI concepts and devices.

15. SUBJECT TERMS
Human-System Interface, Multi-Modal Interface, Voice Recognition, Voice Synthesis, Satellite, Workstation Design

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
Unclassified	Unclassified	Unclassified	Unlimited	569	John D. Ianni
					19b. TELEPHONE NUMBER *(include area code)* (937) 255-8865

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE IS INTENTIONALLY LEFT BLANK

Executive Summary

This report summarizes the development of an improved Human-System Interface (HSI) for space operators, and the development of a testbed for evaluating this and other new HSI concepts. This work was performed under a Small Business Innovation Research (SBIR) contract to Monterey Technologies, Inc. (MTI). The U.S. Air Force Research Laboratory at Wright-Patterson AFB, OH monitored the work. The work was performed between April 2000 and April 2002.

In order to focus the HSI development effort on bottlenecks a task analysis was performed early in the effort. The baseline HSI for this task analysis was the COBRA workstation, which is used in the U.S. Air Force's Center for Research Support. COBRA was selected for this analysis as it is one of the most advanced controller HSIs operated by the Air Force. The results of this task analysis showed that an overly large proportion of the tasks performed by the controllers were done manually and that these tasks were often related to operation of the system. The results also showed that considerable time and effort was spent obtaining the values of variables needed by the controllers to make the required decisions during a support. Based on these results, the effort focused on improving the HSI by (a) shifting some of the task load to modalities other than manual and (b) eliminating unnecessary control actions, and (c) providing information to the controller in ways that were both readily usable and that reduced the demands on the controller's memory.

The HSI developed by MTI is multi-modal. The controller may interact with the system through a conventional mouse and keyboard interface, touch screens, or through a voice I/O system. These interface techniques are redundant in the sense that the controller may use any one of them to accomplish a goal.

In addition to the physical interface, improved display concepts were developed and implemented. The displays were designed to allow a single controller to perform a support. The design of the displays allows the controller ready access to the information needed to make decisions required during a support. For example, if the support plan being followed requires the controller to evaluate a particular variable is within a certain range, then that variable is displayed in close proximity to the step in the plan that instructs the controller to make the comparison. This eliminates the need for the controller to search for the variable of immediate interest, and reduces the need for the controller to put the value into short-term memory in order to make the comparison.

Graphical presentations of the values of variables are also available. These graphical presentations are generally in the form of dial gauges. This format allows presentation of the variable's current value as well as presentation of warning and caution ranges. This format is easily interpretable by the controller, and eliminates any need for a controller to memorize and recall the numeric values of those limits. This reduces the controller's workload.

In order to reduce the requirement for controller training, the values of the data are presented in normalized format by default. This allows controllers to simply know that values near 100% are nominal. This means that controllers supporting a constellation of heterogeneous satellites will not have to memorize the specific values of the variables. This is similar to the manner in which thrust is displayed to pilots flying jet engines. 100% is always maximum continuous thrust regardless of how many pounds of thrust are being generated. Pilots don't need, or want, to know the raw thrust value.

The second thrust of this effort is the development of a HSI testbed. An unclassified simulation of a Defense Satellite Communication System (DSCS) satellite was developed. This simulation includes four major subsystems; Link 1 communications, Link 2 communications, Propulsion, and Electrical. This simulation is of sufficient fidelity to allow investigation of the effects of HSI concepts on human performance. This simulation does not replicate the interactions between satellite components, and is, therefore, of low fidelity.

An experimenter workstation was developed for the testbed. This workstation allows the experimenter to configure and control the simulation. The experimenter has a variety of controls available at the workstation. These include:

- Flow Control (i.e., start, stop, pause, and resume the simulation)
- Data Collection (on, off, and name of the data file)
- Data Collection Rate (0.2, 0.5, 1, 4, and 10 Hz)
- Configuration Selection
- Anomaly activation and removal
- Entry of event markers into the digital data stream
- Entry of Annotations

The experimenter's workstation, like the controller's workstation, allows tactile inputs. This allows the experimenter an alternative to the conventional mouse and keyboard interface. In practice, tactile inputs appear to be particularly useful as a means to enter event markers and for control of the experiment's flow.

Table Of Contents

ADVANCED INTERFACES AND TESTBED FOR SPACE OPERATOR CONSOLES

Background

Maintaining the health of satellites is a labor-intensive task. In United Stated Air Force Satellite Operations Squadrons (SOPS) teams of controllers work together to monitor telemetry received from the satellite and to issue commands that the satellite executes. Contacts with most satellites are performed often – in many cases more than once per day. The frequency with which a satellite is contacted depends on a variety of factors including the type and altitude of orbit, the position of the satellite relative to the earth, sun, and moon, and the general health of the satellite. Other considerations, such as the need to support warfighting and intelligence gathering efforts, also impact the frequency that a satellite must be contacted. The times that contacts can be made are determined by the availability of a remote tracking station (RTS) that can "see" the satellite. Therefore, SOPS generally operate 24 hours a day, 7 days a week.

Satellite controllers, also known as "space operators," play a central role in the operation of the satellites. The controllers contact the satellite, monitor the data from the satellite to identify trends or anomalies that affect the current or future operation of the satellite, and issue the commands to the satellite. In the SOPS, multiple personnel serve as a control team during a satellite support. This redundancy provides a means to verify the accuracy and timing of commands that are issued to the satellite. One of the main drawbacks to this approach is the cost of personnel necessary to make the contacts. These costs include the direct costs incurred performing supports, as well as the training and certification costs needed to insure that all of the control personnel are certified to perform those functions. These indirect costs are significant, and are driven in part by the need to train the controllers to use a poor Human-System Interface (HSI).

VOLUMINOUS AMOUNTS OF DATA

As satellites grow in complexity and as technology provides the capability of offering more and more information, the limits of human information processing can be exceeded with the presentation of the various temperatures, pressures, safety indicators, state vectors, fuel supplies, battery power, and relay states. Indeed, the complexity of a spacecraft – or even a spacecraft's subsystem – is so great that one operator is typically unable to monitor all the telemetry points simultaneously. In some cases, more than 10,000 telemetry points are sent down. The general trend towards less manning only exacerbates the problem. Accordingly, the necessity for well-designed displays is more important now than ever before.

HIGH COGNITIVE DEMANDS

Understanding the cognitively demanding nature of the Space Operator's work emphasizes the importance of well-designed console displays. The Space Operator work is much like that of an aircraft pilot. Their tasks are characterized by "complexity, great size (both physical and conceptual), the presence of risk and high levels of hazard, both low and high levels of operator

workload, complex real-time dynamics, and a need to integrate the activities of many people. There can be very high levels of automation, and very high implicit or explicit costs in terms of performance and safety" (Moray, 1997). There is however one significant difference. As the control room operator on duty during the infamous Three Mile Island accident, Edward Frederick, put it,

> *"In an airplane, you can look out the window and see where the mountains are, you can see whether you're over water or land, you can feel the attitude of the plane, you can hear noises from the engines, that sort of thing. In a control room, the environment is quite opaque, you can't see through it. All you have is the instruments. You can't hear the noises, you can't hear the engines running, you can't feel it moving. You have to rely on the instrumentation to show it to you. So the extent to which we rely on instrumentation is probably many times greater than an airline pilot or a train engineer."* (Edward Frederick as quoted in Maddux and Muto, 1999).

The significant cognitive demand created by having all relevant information presented solely on instruments becomes clear when one relates it to the characteristics of human decision making behavior. The most advanced and accepted model of human decision making contains three levels of behavior (Rasmussen, 1980). Rasmussen calls these levels skill-based, rule-based, and knowledge-based.

Skill based behaviors occur when a sensory input triggers a subconsciously generated response. An example of a "... skill-based behavior occurs when an aircraft banks left and the pilot's response is to immediately bank the aircraft to the right to achieve a wings level attitude (Reising and Aretz, 1987).

"Rule-based behavior occurs when a signal or series of signals is recognized in memory as applying to a specific set of rules." (Reising and Aretz, 1987). An example of rule-based behavior occurs when an emergency indicator appears on the display and the operator carries out the emergency procedure rules.

Knowledge-based behavior is defined as an action that is applied to situations where no prior set of rules exists, but must be created. In knowledge-based behavior, the various sensory inputs first have to be synthesized in order to formulate new rules. One the problem is formulated in this manner, "... the human resorts to rule-based behavior and applies the appropriate actions indicated by the rules" (Reising and Aretz, 1987). An aviation example of this type of behavior is a "... pilot becoming aware on an unexpected threat to his aircraft, such as a surface-to-air missile site. The first thing the pilot has to do is to formulate the problem by gathering data from his displays" (Reising and Aretz, 1987). In this example, the problem formulation or synthesizing process involves "... assessing the lethal potential of the threat site and determining the appropriate tactical strategy (avoid or destroy the threat) to handle the situation. Once the problem is formulated, it can be solved by applying the appropriate rules" (Reising and Aretz, 1987).

Skill-based behaviors are the easiest. Satellite operator's work is in the realm of rule-based and knowledge-based behavior rather than skill-based. In addition, "... most problem solving situations require the formulation of new rules. Unfortunately, humans are poor at processing data from several sources and making decisions" (Reising and Aretz, 1987). In fact, human decision making behavior is best in situations containing moderate levels of complexity. Therefore, the ideal human-system interface (HSI) would allow the operator to work in a rule-based behavior environment as much as possible. :If data are preprocessed and presented properly, rule-based behavior can be applied to a knowledge-based problem – but the key is that the information presented must aid in problem formulation" (Reising and Aretz, 1987). The benefits of a well-designed display can be realized in the problem formulation process. With no direct sensory input available to the controllers and with so much knowledge-based decision making required, the benefits to be gained by "properly" presenting the information are significant.

ANTIQUATED DISPLAYS

The displays used in most satellite operations squadrons are outdated. For example, the displays used by the operators in the 50th Space Wing were designed in the 1970s and installed in the 1980s. Figure 1 is an example of one screen used in the 50th Space Wing. In a survey of the operators performed during Phase 1 of this program, the displays and equipment were described as "archaic," "severely outdated," "antiquated," and "obsolete". It is unreasonable to expect maximum effectiveness from operators using this level of technology.

ASOFT	THRS	BODY AXIS DELTA		CST	2.2996402344E+04
- - - - -	- - - - -	R	1.0038E+00	CX	-2.389046000E+07
ASFAT	3.3117E+02	P	1.0835E-03	CY	1.3667987200E+08
ASMAG	7.6572E+01	Y	2.5163E-03	CZ	8.3318875000E+05
ASDIR	0.0000E+00			CXD	-9.826904297E+03
APVSF1	-3.944E+01			CYD	-1.709321289E+03
APVSF2	6.5351E+01	DELTA VEL BODY		CZD	-5.536420898E+02
		X	-2.345E+02		
		Y	1.1017E+02		
GSOFT	THRS	Z	7.8458E+01	PMT	2.3370707031E+04
- - - - -	- - - - -			PX	-2.755944800E+07
GSFAT	2.2895E+00			PY	1.3598966400E+08
GSMAG	2.8187E-01	ATT RATE FILTER		PZ	6.2567587500E+05
GSDIR	0.0000E+00	R	1.0036E+00	PXD	9.857171875E+03
GPVSF1	-2.794E-01	P	1.0556E-03	PYD	-1.995220215E+03
GPVSF2	-5.183E-02	Y	2.5814E-03	PZD	-5.251048584E+02

Figure 1. Example of a telemetry screen.

Clearly, the information formatting in the example above is not conducive to situational awareness. In addition to the lack of color coding and graphics, there is virtually no forethought into the presentation and organization of the data. And of course, there is no utilization of any modern technological console advancements. The display is presented much like a phone book – only without alphabetic organization.

The current operator consoles not only lack near-future technologies, but also current technologies – now available as commercial off-the-shelf (COTS) prducts. Achieving maximum benefits from incorporating near-future technologies requires that these advanced technologies be incorporated into a relatively modern platform – not a 20 year old system.

There are several on-going satellite control center modernization programs aimed at decreasing staffing requirements, lowering training costs, reducing support costs, and improving operator performance. Most of the current modernization programs are aimed at making better use of today's technologies, including currently available human-system interface (HSI) technologies such as Windows and web-bases graphical user interfaces (GUI). While well designed GUIs will result in worthwhile improvements in the near-term, new performance-aiding technologies will be required to meet the increasing demands related to future constellations of satellites and reducing staffing levels.

Space operators are not realizing any of the benefits from new HSI technologies such as multi-sensory displays, multi-modal controls including natural language input, intelligent decision aids, and immersive interfaces. Research on these technologies in Unmanned Autonomous Vehicles (UAV), shipboard watch stations, and futire command posts has demonstrated that they can significantly reduce manning, training, and support costs while simultaneously improving operational performance. Brief descriptions of some of these technologies and their benefits follow:

- *Multi-sensory display* of information enhances operator situation awareness and performance by reducing visual workload, augmenting limited visual display area, and enhancing system alerting and operator guidance. The use of synthetic spatially oriented 3D audio to provide multiple auditory sources and directional alerts could prove effective in alerting and directing the attention of space console operators to off-screen events.

- *Multi-modal controls* improve operator performance, reduce training time, and reduce operator cognitive workload. In particular, spoken language interfaces, combining new speech recognition and natural language interface technologies, are beginning to be deployed to enhance operator performance and reduce training in a variety of commercial and military operational settings (including command and control centers very similar to space command centers).

- *Intelligent decision aids* improve operator performance by reducing the need to cull through large amounts of data to extract information to achieve understanding, enhance the quality of operator decisions, reduce the effects of stress, and reduce the need for support personnel thus facilitating reduced staffing/

None of these technologies are currently being used by the 50th space wing operators. Consequently, none of the corresponding benefits are being realized.

PROGRAM GOALS

This Small Business Innovation Research (SBIR) program (not to be confused with the Space Based Infrared System program which has a similar acronym) is aimed at addressing the HSI issue in the satellite control environment. The two major goals of this program are:

1. Develop an improved HSI for use by satellite controllers
2. Develop a testbed in which HSI concepts can be evaluated.

Both of these goals have been accomplished. The improved HSI has been built into the testbed.

Existing Controller Workstations

SATELLITE OPERATIONS SQUADRONS.

The controller workstations in the SOPS are legacy systems. The HSI typically consists of a display, a keyboard, and possibly a mouse. The data from the satellite (i.e., the measurands) are presented in alphanumeric format. With the exception of rudimentary color-coding of the alphanumerics, display techniques designed to enhance the controller's ability to perform the tasks required during nominal and off-nominal supports are not generally available.

The data from the satellite are checked and monitored by the controllers during a support. Controllers are expected to be able to detect anomalies and identify trends in the data based on their examination of these data. Identification of trends allows the controller to anticipate the future state of the satellite and to determine the proper action to perform. Knowledge of trends also allows the controllers to perform defensive actions to counter attacks. The data are presented as alphanumerics that can change at a rate of 2 Hz. In order to detect trends, the controller must build a mental model of the pattern of changes for a particular variable over the course of the support, or over supports performed over days or weeks for trends with extended time courses. The data presentation formats do not support controller performance of these tasks as well as other display formats might which put the data into a context that reduces the memory load placed on the controller.

The SOPS rely heavily on paper products coupled with the knowledge of the personnel on the controller team to determine the correct commands to enter in during a support. A significant part of the control task is verification of the commands entered at the workstations. This is necessary because the consequences of sending an incorrect command can, in the worst cases, cause the satellite to be lost.

CERES

The Center For Research Support (CERES), located at Shriever AFB, Colorado, has developed and is using a different approach to performing satellite supports than those used by the SOPs. At CERES a single controller performs a support. The commands that are issued during a support are contained "pass plans." These pass plans are prepared prior to a support and contain the commands that will be up-linked to the satellite. This approach allows the commands to be

prepared and checked for accuracy (e.g., this eliminates the possibility of an entry error) and sequence before the satellite is contacted. Pass plans that perform different functions are usually performed in a predetermined order during a support. This sequence is defined in the Program Action Plan (PAP) for that support. The PAP is usually developed by the satellite engineer and executed by a controller.

Pass plans that are not contained in the PAP for that support are also available for the controller to use. A pass plan to perform an action that the controller deems necessary based on inspection of the satellite data can be brought up from a library of pass plans. For example, if the controller noted that a pressure was nearing a lower limit, then the pass plan for pumping up that pressure would be called up and executed, even though that aspect of the support was not contained in the PAP. In some cases, controllers are required to get approval from the cognizant satellite engineer, or even to have the satellite engineer present at the workstation, in order to perform some pass plans or steps within those pass plans.

The HSI at CERES is the Commercial Off-The-Shelf (COTS) Based Real Time Architecture (COBRA). COBRA is a major step forward from the workstations used in the SOPS. However, it is not without shortcomings that adversely effect the controller's ability to perform a support. Like the controller workstations in the SOPS, the workstation in CERES is a keyboard and mouse HSI. Similarly, the data down linked from the satellite is not presented in a way that supports controllers. The data is poorly organized and presented in alphanumeric format with mnemonics (cryptic field descriptions) and "raw" data values. Furthermore, the data is located on a separate display from the pass plan. This setup requires the controller to use a separate mice and keyboards for interacting with each of the data and pass plan displays.

While CERES has advanced the state of the art in satellite support techniques, the presentation of raw measurands is not an improvement over the presentations of similar data in the SOPS. The measurands are still presented in a "phone book" format. In the COBRA interface, there is a provision for color coding of the data, but color coding is not used consistently so its value to the controllers is compromised. Because of the inconsistency, the controllers have learned to ignore the color coding, which defeats the purpose. Controllers in CERES also examine screens of alphanumeric data.

Future Concepts of Operation

Alternative Concept of Operations

Outside of the Air Force, there is a trend towards unattended or autonomous operations. NASA and the University of California operated the Ultraviolet Explorer using this concept to reduce the manpower costs associated with supports. Human intervention was required only when unanticipated events that the satellite was unable to resolve. We anticipate that this type of operation will become more common with military satellites. However, because military satellites are often re-tasked and because they are subject to attack, human interaction with these satellites is likely to remain a requirement.

Increased use of unattended operations is likely to replace routine state of health assessment supports at some point in the future. The satellite will then alert the control facility when there is an unresolved problem. This approach will reduce the life cycle costs of supporting satellites.

The unattended operations concept brings in a number of potential problems. Two examples of these problems are:

- Controller lack of familiarity with the satellite
- Controller lack of familiarity with the workstation

Satellite Familiarity. The problem that first comes to mind is the lack of familiarity that the controller is likely to have with the satellite. In the case of unattended operations, a controller will perform a support when the satellite has detected and reported a problem it cannot resolve without human intervention. In essence, the satellite will "phone home" when there is a problem. In this case, the controllers will not have the familiarity with the expected values of the measurands for that satellite that they would if supports were performed frequently. This will be compounded by the likelihood that the controller will be supporting multiple, possibly heterogeneous, satellites simultaneously. This opens the door to the possibility of negative transfer between the satellites. As an example, the controller could easily confuse the end points of the nominal range of a variable on one satellite with those of another satellite ("Was that limit 23.2 volts or 22.3 volts?") possibly with a disastrous effect.

The second, related problem is that the controller will not observe the satellite over an extended period of time. This will reduce the controller's knowledge of the factors that have lead to the anomalous condition and reduce the controller's ability to predict the results of their actions. (This is now often considered as part of the construct known as situation awareness [SA]). For example, the controller will not know if the observed problems have occurred gradually or abruptly. One can imagine any number of scenarios where the actions performed by a controller would differ depending on the type of onset for a particular problem.

Workstation Familiarity. Since controllers will not likely be performing supports as often as they do now, it is easy to predict that controllers will not be as effective or efficient using their workstation(s) as they are now. One solution might be to have the controllers perform simulated routine supports in order to maintain their proficiency. This has the obvious drawback of requiring all the work and costs associated with performing those supports, as well as requiring simulations of the satellite and all of the systems the controller uses to perform a support. It isn't difficult to judge this approach to be a "non-starter."

This isn't to say that training and simulation won't have a place in the future. The point we are making is that the routine supports now allow the controller to observe the satellite over an extended period, and to maintain a high degree of familiarity with the capabilities of the workstation. That is, each support provides the controller a small amount of recurring, hands-on training. In the event that routine supports are not performed manually, then this training will not be available and the readiness of the operators will decline unless other measures are taken. These other measures may include scenario based training focusing on controller responses to anticipated conditions outside the capability of the autonomous system. Controllers would also

use scenario based training to prepare to counter attacks mounted against one or more satellites in the constellation.

A better approach is to design a workstation HSI that allows the controller to interact with the satellite in a natural, intuitive manner. That is not to say that the knowledge of the controllers about the satellites can be lowered. When the satellite "phones home" for help with a condition it cannot resolve autonomously it means that a person very knowledgeable about the satellite is needed. Otherwise the system would have taken care of the problem using its on-board knowledge and capabilities. Instead, it means that the knowledge that the controllers will need about the workstation and its ancillary equipment would be lowered so that the controller, or satellite engineer, can focus on the solving the satellite's problems.

At best, reduced familiarity with the workstation or with the satellite would mean that supports will be longer, thereby tying up capacity of the AFSCN assets. At worst, it would mean that the controller may not be able to identify a problem, determine the remedial action to be taken, and to initiate those actions in time to save the satellite.

Controller Task Analysis

In order to identify areas where the current HSI impedes controllers, a task analysis was performed. The results of this effort were used as inputs to the design of the improved operator workstation. Specifically, we attempted to develop and incorporate solutions to the bottlenecks identified as part of the task analysis. The following are highlights of that report. The entire report containing more detailed discussions of the methods, results, and recommendations is contained in Appendix 1.

METHODOLOGY.

Experienced satellite controllers from CERES performed ten scenarios using the COBRA workstation. The screens viewed by the controllers during each scenario were videotaped for later analysis. A verbal protocol was used. In this protocol, the controller describes aloud what they are doing and why. The analyst was able to make inquires during the process in order to obtain clarification. This approach allows the research team to catalog the actions performed by controllers. The information sought by the controller (i.e., the data that they are interested in at that point in the scenario) is also captured using this process.

Because the controller is verbalizing the action being performed, the time data that one could extract from scenarios using the verbal protocol is unreliable. If the controller has a lot to say, or if the analyst has questions, then the time course of the scenario will be delayed. In this case, the lack of time data was considered to be acceptable because (a) the insights gained from the controller's verbalizations were expected to be very useful in the development of our understanding of the tasks and (b) the available simulation at CERES was known to be incomplete in some aspects. This lack of completeness was expected to make it impossible to perform some of the tasks in the scenarios at the same tempo as they would be performing a support for an actual satellite. In order to address this shortcoming in the data obtained in the task analysis, an experienced researcher provided time estimates for many of the tasks. These time estimates should be considered ordinal data, rather than as interval data.

RESULTS AND RECOMMENDATIONS.

A total of 921 individual tasks performed by the controllers were identified. The modality or modalities used by the controller to perform each task were identified. Figure 2 shows the proportion of tasks in which each modality was employed.

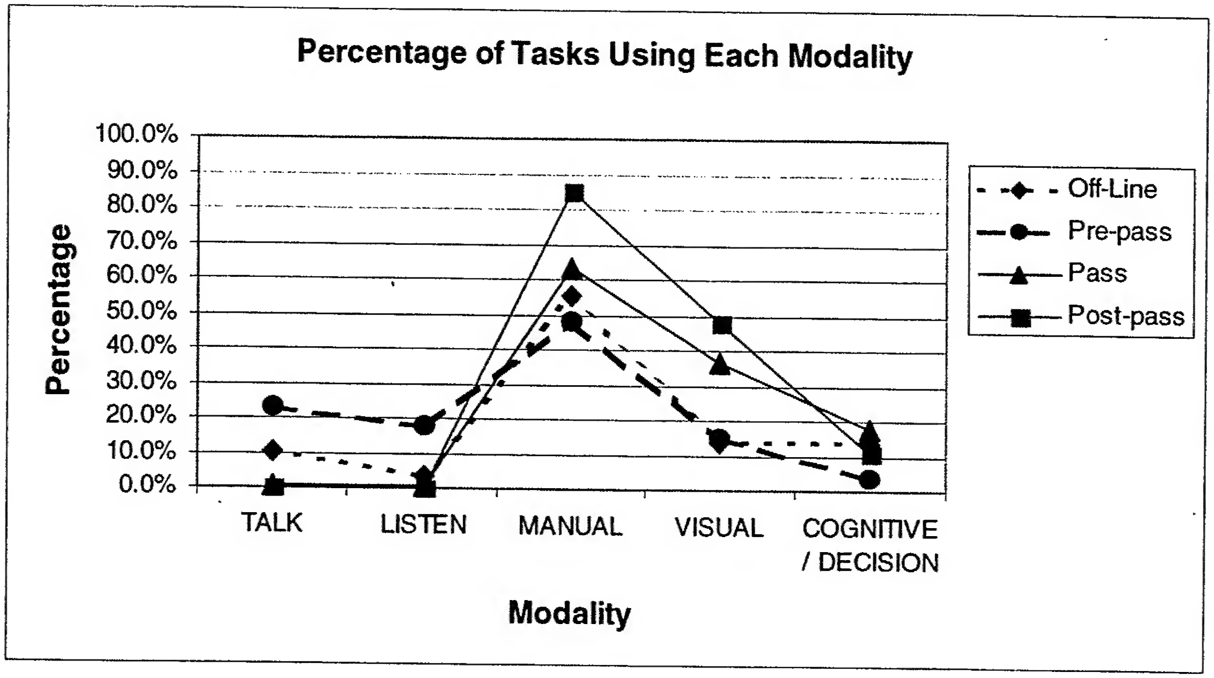


Figure 2. Proportion of tasks in which the controller talked, listened, performed a manual task, performed a visual task, or performed a cognitive/decision-making task.

<u>"Manual" And "Visual" Modality Loadings</u>. Examination of this figure shows that the "manual" and "visual" modalities are most heavily loaded. Looking at the tasks themselves it is apparent that this heavy loading is due to the controller interacting with the system via the mouse, which requires involvement of both the "visual" and "manual" modalities. Further inspection shows that many of the mouse actions are due to a requirement to click three or even four times to execute a step in the pass plan.

The load on the "visual" modality is compounded by the requirement that the controller visually search for and read the data value of interest.

This figure also shows that during and after the pass the talk and listen modalities are essentially unused. This suggests designing an HSI that allows the controller to use these modalities to command the system and to obtain information from the system.

Two methods of reducing the controller's manual load are rather obvious. First, eliminate the requirement for multiple mouse clicks. Most commands should be executed with a single click.

Multiple clicks should be required only when the intent to execute a command must be verified. In these cases, the design should not allow the controller to "mindlessly" double click. Here some other action should be required, such as repositioning the mouse on a specific button before the second click can cause an action to occur.

The second approach to reducing the dependence on the manual modality is to provide the controller another means to issue commands. In light of the under-utilization of the voice modality, a voice interface for commanding the system and for receiving information from the system is an obvious candidate.

The design of the COBRA workstation requires the controller to use two mice and two keyboards during a support. The mouse and keyboard on the left are used in conjunction with the left display, and the right ones with the right display. The ease of use of the workstation would be improved by allowing a single set of control devices to interact with all of the display surfaces.

Facilitate Comparisons. A large number of tasks require the controllers to compare an observed value with an expected or allowable range. Often the observed value and the range information are physically separated, possibly on separate displays. Thus the controller must place either the current value or the range information into short-term memory (or, perhaps to recall the range information from long-term memory) before comparing the values. Placing the values in proximity would lessen the time required to scan for the needed information and, more importantly, reduce the load on the controller's memory thereby reducing the probability of an error. Simply placing the current value in the pass plan adjacent to the step that instructs the controller to make a comparison would improve this aspect of the HSI.

Along these same lines, the HSI would be improved if the controller was able to determine whether or not a variable is within tolerance without requiring any mental comparisons. One HSI approach that addresses this issue is to present a graphical indication of the variable and the allowable range of values. Dial gauges in which the thickness and color of the fixed scale are used to show the nominal, warning, and caution ranges is one such approach. Other graphical formats may also be used with good effect.

Design of Controller HSI

DISPLAYS

Pass Plans. Pass plan displays were developed based on those in use at CERES. These were changed in several ways to overcome the problems identified from the results of the task analysis. These modifications include:

- Locating data needed by the controller in order to make decisions in the pass plan adjacent to the steps that require that information, rather than having the data embedded with other data on a separate display. This eliminates the need for the controller to search for the data and makes it easier to compare the value with the range called out in the pass plan.
- Reducing the number of mouse clicks required to execute the steps in the pass plan. In order to execute each step in the pass plan at CERES, controllers are required to click the button on

the mouse three, and sometimes four times per step. This is a legacy of the "bad old days" when the controllers were expected to re-check their typing of the command before sending the command to the satellite. As the pass plans contain commands whose accuracy has been verified, there is no need to verify the intent to send the command.

Subsystem Diagrams. Subsystem diagrams have been implemented for the Link 1, Link 2, and Electrical subsystems. These diagrams are based on a concept developed by Mr. Chad Oster of CERES. These diagrams give the controller a schematic view of each subsystem. These schematics show the flows through the system, the states of the discrete variables, and the values of the continuous data being received from the satellite. Figure 3 is an example of one such diagram.

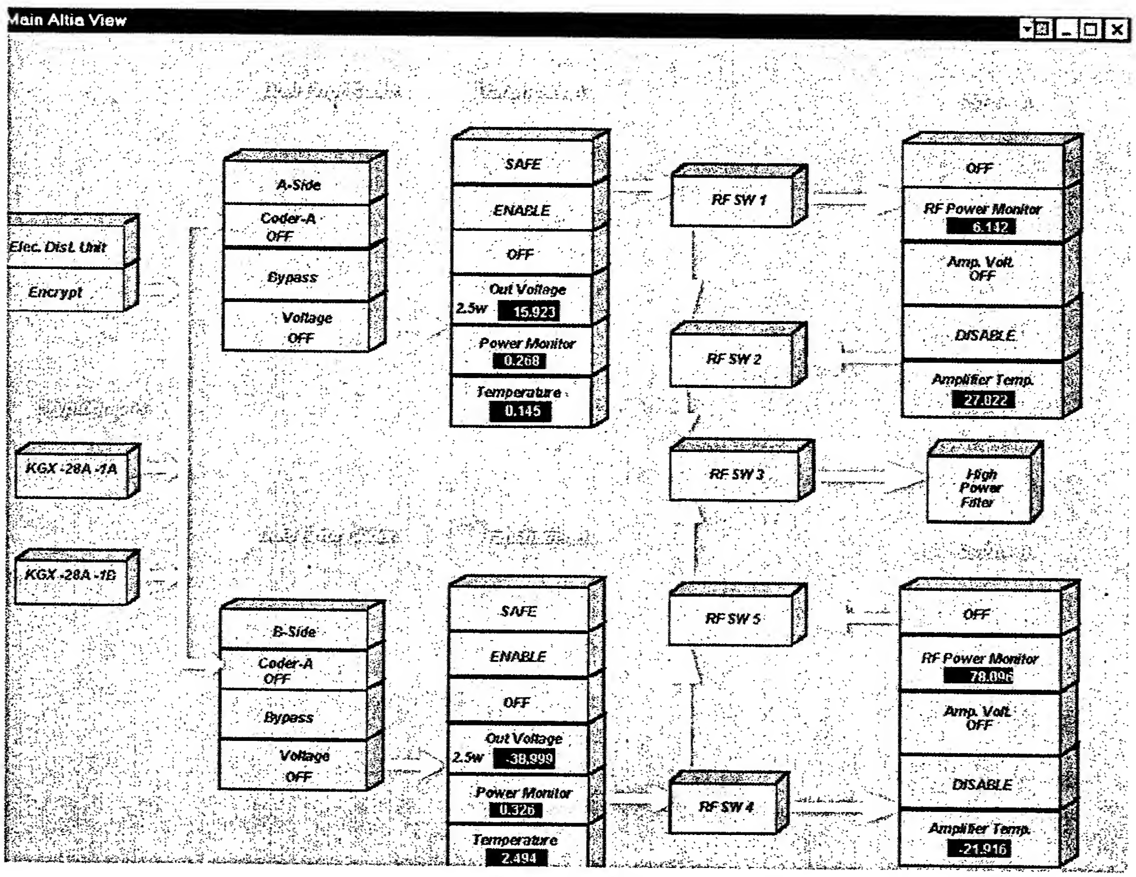


Figure 3. Link 1 Communications Subsystem Diagram.

One advantage of this display format is that the controller is able to see the entire subsystem as a whole. This allows the controller to visualize the interrelationships between components and better anticipate the effects of control inputs.

Use of a subsystem diagram opens up new uses for a touch interface. Here, the controller can press on a button to change the state of the variable. For example, in Figure 2 the A-side Coder-A is OFF as seen in the upper left quarter, in the column labeled "Dual Error Coder." This HSI allows the controller to press on this button to turn the device ON. This would be represented by

the button appearing to be depressed, and the text inside the button changing to reflect the ON state. In the event that the switch was unable to move to the ON position aboard the satellite, then the controller would see the button depress momentarily and then return to the OFF position. (Changing a switch position might be impossible for a variety of reasons. For example, the switch or its actuator might be physically damaged.) The momentary change of state provides the controller an indication that the HSI accepted and attempted to execute the command. The return of the button to its original position shows the controller that (a) the satellite was unable to perform the request and (b) the actual state of the satellite. This information is needed in order for the controller to develop a plan to accomplish the objective of that support.

Continuous Monitoring of Data. An indication is presented if there are any warnings or cautions. There is a positive indication if all is well. This allows the controller to examine a single location on the workstation to determine the current status of the satellite. If all is well, a green circle is displayed. If there are one or more warnings, a yellow square is displayed and the green circle is removed. If there are one or more cautions, then a red triangle is displayed and the green circle is removed from the display. The warning and caution indicators are displayed near, but not at the same exact position as would be the green circle. This provides redundant shape, color, and position coding for this information.

Voice Synthesis. Voice synthesis is used to provide information to the controller. This information is in two general categories. First, if the controller inquires about the current value of a particular variable (e.g., "tell me the temperature of the coolant") then the system will announce the value ("the temperature of the coolant is 87.3 degrees"). This allows the controller to get access to a value when his visual system is otherwise occupied.

During evaluations of the voice synthesis system, it was found that full sentence responses were judged to be too verbose. In the case of humans working together as a team, it is understood that the value being reported is the one requested. More specifically, too much time elapsed from when the system began speaking and the key information was delivered to the controller. For example, if the system said, "the temperature of the coolant is 87.3 degrees" the phrase "the temperature of the coolant is..." was of much lower value than the phrase containing the temperature. Users preferred responses that contained only the key information (e.g., "87.3 degrees"). This preference was reported regardless of whether the numeric value preceded or followed the descriptive phrase.

This preference is not surprising in a system trusted by the user. However, it was expected that controllers would need to use the system for awhile before developing this level of trust. It may be that the controllers who used are system, who are used to dealing with advanced HSI systems, have more trust in this level of technology than do controllers who have not worked as extensively with this level of HSI technology. It may also be that this result is an artifact caused by the fact that the voice recognition and synthesis processes needed to be completed by the system before the controller could move on to their next interaction with the workstation. Additional research would need to be conducted in order to fully understand the reasons behind this preference.

The second type of voice synthesis is confirmation of controller requested actions. When the controller instructs the system to make a change in the setting of a discrete variable then the system will respond by verbally confirming that the change has been made. For example, if the controller says "turn on the heater" the system would turn the heater from the off to the on position and respond "heater on". The response provides the controller a positive indication that the requested change was made. This is similar to what happens now when controllers and satellite engineers team to perform a support and provide verbal confirmation that the requested action has been performed. As in the case of continuous variables, evaluations showed that users preferred shorter verbal confirmations over more complete ones in which the name of the component is included.

<u>Data Presentation</u>. HSI shortcomings in the method and format of data presented to satellite controllers in the workstations in CERES and the SOPS include:

- Color coding not used effectively
- No context is provided
- Poor Organization of data
- Selection of Units of Measure

Color-coding. An attempt is made to use color-coding in CERES and in the SOPS. However, due to resource limitations color-coding is all too often not up to date. When the color-coding is not kept current, the controller cannot rely on this display dimension to indicate the presence or absence or a problem. This may be a bigger problem in CERES due to the ages and conditions of the Track and Control Only (TACO) satellites they control. For instance, if a TACO satellite is out of propellant for a particular thruster, then that value might be displayed in red on a screen. However, because that is now the NORMAL condition for that satellite, the controllers learn to ignore the value in red because "that's always red." The controllers quickly learn that they can't rely on color to show them a problem, so they effectively ignore the color-coding. Clearly, for color-coding to be effective it has to be accurate and up to date. Maintaining the currency of color-coding requires the application of some resources. This is an organizational issue, and beyond the scope of this effort.

Assuming that the limits for each variables nominal, warning, and caution ranges are kept up to date, color-coding can be a very useful tool in allowing the controller to detect off-nominal situations. In this HSI, color-coding is used in a redundant fashion with other dimensions. For example, a continuous monitor shows the controller at a single glance if one or more values are in the warning or caution ranges. This indication combines color-coding with shape and position coding. In this case, if all variables are in their nominal ranges, then a green circle is displayed prominently. If one or more variables are outside the nominal range then the green circle is removed from the display and a yellow square or a red triangle (or both depending on the values of the variables) are displayed in near by locations. These locations are fixed, allowing the controller to develop and use a consistent scan pattern.

One of the design principals here is that color-coding is not sufficient, by itself. Instead, color-coding is best employed where the same information is coded in a different dimension. This approach allows the code to be understood if color becomes unavailable (in the "bad old days", for instance, one or more of the guns in a CRT could fail). This approach recognizes that while

military satellite controllers are currently required to have normal color vision, controllers in commercial settings do not have to meet this requirement. Furthermore, in the future it may be that other personnel, such as satellite engineers or orbit analysts, may participate more directly in the support. These individuals are expected to have the same distribution of color deficiencies as found in the general population.

Another design principal incorporated into the testbed HSI is the use of positive feedback. One of the first issues that comes up in the design of an HSI is whether or not the absence of an anomaly or out of tolerance indicator is sufficient, or if the system needs to explicitly inform the user that all is well.

A HSI without positive indications often leaves the user uncertain as to the state of the system. Is the system operating as it should? Or is there a problem and the display isn't operating correctly? Imagine if traffic signals had only red and yellow lights. As you approach an intersection where neither of these are illuminated, are you supposed to proceed through the intersection, or are the red and yellow lights burned out? Clearly, a green signal light, which is a positive indication, makes it less demanding to drive. This uncertainty places the demand to verify the system's state back on the user. This increases the load placed on the user, and reduces the usefulness of an automated continuous monitor of the system.

Serendipitously, a positive indicator provides an opportunity to use position coding to help make changes in state more conspicuous, or attention grabbing, to the user. When a light changes position and color it can be quite conspicuous, reducing detection time. In this context, it may be more important to let the controller know that a problem has been fixed in a timely manner than to cue them that something is amiss, but the principal is the same.

In a mission critical system such as a satellite control workstation we believe that a positive indication is required.

Lack of Context. In terms of no context, the problem is that the controller must recall the warning and caution limits for the variable. This is, clearly, a difficult memory tasks for any one satellite, and is only going to be worse when flying multiple satellites of different types. This problem is, we believe, one of the major contributors to the need for initial and recurrent training of controllers.

We have addressed this issue by displaying the data values in graphical form. These graphs, which are dial gauges in this implementation, clearly indicate the warning and caution ranges for that particular variable. The controller is able to determine if a value is out of tolerance visually; they do not have to recall any values from memory, nor do they have to perform mathematical comparisons between the current value of the variable and its limits.

Controllers, by virtue of performing supports on a satellite over an extended period of time, are familiar with the ranges of values for many of the measurands and the trends in the way those values change. This frequent interaction with the satellite allows the controllers to see trends in the changes of the measurands. For example, a controller might know that a voltage tends to be on the low side of the normal range at a particular point in time because of the relationship

between the satellite, earth, and the sun. Once that relationship changes, then the voltage is expected to drift back towards the center or upper portion of the nominal range. In other words, the frequent interactions with a satellite the controller becomes knowledgeable about the satellite and the patterns of changes expected in the data.

As unattended operations become more prevalent in the future, controllers will not be getting the same exposure to the satellite, and won't have the same awareness of the situation. To further provide a context for the value of a variable, the controller may have the current and past values of a variable displayed as a histogram. The aim of this presentation mode is to allow the controller to see the trend of the values over time. Trend information can be invaluable in determining the cause of an anomaly, and in developing an approach to restore the satellite to a nominal operational condition. For example, the action a controller would take is expected to be very different if a voltage drops slowly over an extended period of time, compared to a case where the voltage abruptly drops. In the former case, the controller might focus on the condition of batteries aboard the spacecraft or on the configuration of the charging system. In the latter case, a controller might suspect damage from an acute event to a solar cell or other system component, possibly from a collision or from intentional acts.

Data Organization. The organization of data values on the workstations in the SOPS and in CERES is not always as intuitive as it could be. In some cases, a single variable will be listed in multiple locations. This can increase the difficultly of finding the value of the variable of interest. The controllers must memorize the data page or pages on which the variable appears, and the location of the variable on each page.

We have addressed this problem by (1) co-locating the value of the variable, be it a discrete state or a continuous variable, in the pass plan where it is needed by the controller, and (2) by including the values directly in the schematic diagram of that subsystem.

By placing the value in the pass plan, the need for the controller to search for the variable is entirely eliminated. Thus the controller would not be required to hunt down the relevant data. This reduces the controller's memory load and the need for manual actions.

A more meaningful organizational scheme can be provided to the controller by putting the value into the subsystem schematic at the proper location relative to the other subsystem components provides. This scheme is likely to be similar across satellites because while the details of the subsystem will differ from satellite to satellite, there will be some commonality.

Units of Measure. Satellite data is presented to the controllers as actual values. Temperatures are reported in degrees, voltages in volts, currents in watts, and so on. This approach requires the controller to memorize the nominal values for each satellite being flown, and the allowable ranges for those variables. Errors caused by recalling a similar, but incorrect value are more likely using raw values than when using nominal values. For example, it is easy to imagine that a controller might recall a nominal values as being 23.2 volts when the correct value is 22.3 volts. The likelihood of this type of error occurring is increased when the controller flies a constellation of heterogeneous satellites. This problem is addressed by presenting normalized values by default. This means that a value of 100% is always nominal, no matter what the actual

value is or what satellite it being supported. This is similar to the way in which thrust of a jet engine is reported. It doesn't matter what engine is being flown, 100% is always maximum continuous thrust. The pilot doesn't need to know whether that corresponds to 2400 lbs. of thrust or 24000 lbs. of thrust.

Should the controller need the actual value, then a single toggle converts all of the data from normalized to raw values.

CONTROLS

Mouse and Keyboard. A mouse and keyboard interface is provided for the controller's use. Unlike existing systems (i.e., the COBRA system at CERES) a single mouse and keyboard provide access to all of the display surfaces in the workstation. That is, unlike the COBRA workstation the controller does not need to switch control devices when switching focus from one screen surface to another. Instead, the boundaries between the screens are transparent to control actions as if all of the screens are joined and acting as one large screen.

Touch Screen. A touch screen interface is provided in addition to the mouse interface. This interface allows the controller to simply touch the desired location on the screen and the effect will be the same as performing a mouse click at that same location.

In the past, the satellite control community has experimented with touch screens. It is our understanding that touch screens were not favorably assessed. The problems seem to have stemmed from (a) long response latency and (b) inadequate precision.

The first problem, excessive latency, appears when the controller presses the screen but does not receive any feedback that the input was registered by the system. Consequently, the controller presses the screen a second time invoking an unintended input which appears after the system catches up with the first input. This is akin to what happens when the control latencies in an aircraft are poor. This can lead to pilot induced oscillation – an undulating flight path brought about by over-controlling. The poor temporal dynamics of these earlier touch screens frustrate the users at best, and at worst force the user to figure out what state the system is in, and then correct that state. The latter can significantly increase physical and cognitive workloads.

This problem is not evident in the touch screens used in this program. The latency of modern COTS touch screens such as the ones incorporated in this testbed is short enough that the controller receives feedback within an interval that eliminates the multiple touch problem.

The second problem, a lack of precision, stems from attempting to use a touch screen with target areas that are too small, or have no dead band between areas. This would occur, for example, if a touch screen was simply added to a display without adjusting the size and placement of the "virtual buttons" to account for the difference in precision between a mouse and the finger.

We have addressed this potential problem by making the size of the touch sensitive areas appreciably larger than would be the case in the absence of a touch interface, and by separating touch sensitive areas with dead bands.

Voice Recognition. A voice recognition system has been incorporated into the testbed. An extensive set of phrases that are recognized by the system has been created. The large set of phrases approximates a natural language recognition system, as opposed to systems that require a very constrained syntax.

To aid the controllers in making their intent clear to the system, we have employed the following conventions:

- Phrase that begin with "Tell me..." cause the system to announce the value of the continuous variable using the speech synthesis capability
- Phrases that begin with "Show me ..." display a graph showing the value of the variable.
- Phrases that begin with "What is..." and phrases that state only the variable name (e.g., "C+150V" or "SLOART") cause the system to both announce the current value and display a graph.

Although use of these conventions seems stilted at first, in practice we have found that they quickly become incorporated into the user's vocabulary.

Appendix 2 contains a listing of the phrases that are recognized.

In the event that a voice command is detected, but the phrase is not recognized, the system generates a synthesized voice statement indicating that the command was not understood. This feedback tells the controller that the voice recognition system is functioning, but that for some reason the command was not processed and will not, therefore, be executed. The controller then has the option to restate the command as is, rephrase the command, or use another method to enter the command.

When a command is recognized, the system executes that command. The system's response depends on the nature of the command. If the command is a query as to the value of a specific variable then the system responds by announcing and/or displaying the value. If the command instructs the system to change the state of a control, then the switch state is set to the state described in the command and the system provides a verbal confirmation of the change. If the display shows the switch position then the display is also updated. Finally, if the command is to identify all variables in the caution or warning state, then the system displays those variables. The variables are displayed in graphical form if there are six or fewer in the designated state, and as a list if there are more than six.

Design of Controller HSI Testbed

The design of the testbed has evolved during the course of this work. Throughout the development we have attempted to maintain a focus on the ability to test HSI features. Incorporation of a high fidelity satellite simulation while desirable was not, in and of itself, one of our primary goals.

The path to development of the testbed was not without some detours. Our initial aim was to incorporate a high fidelity simulation of a satellite, ideally one supported by CERES. This would be useful because (a) there is an already highly trained cadre of experienced controllers available to participate in Air Force sponsored and (b) research performed in the testbed could be replicated using the track and command only (TACO) satellites supported by CERES.

KEY SYSTEMS - PROBLEMS AND CHALLENGES ENCOUNTERED

Satellite Simulation. Early in the program, MTI and Braxton Technologies (Pleasanton, CA) reached an agreement to cooperate. MTI would provide results of the research to Braxton, give Braxton the right of first refusal on new HSI designs, and would fund support received from Braxton related to the integration of a satellite simulation into the testbed. Finally, MTI would purchase Commercial Off-The-Shelf (COTS) software used in the testbed from Braxton. Braxton, in turn, was to provide COTS software on loan during the development of the testbed, and provide assistance to MTI in integrating their software into the testbed.

A number of technical, financial, and schedule hurdles to the successful integration of a high fidelity satellite simulation emerged. These have been described in periodic progress reports and other communications to the Government's technical representative so are not repeated here.

We would be remiss if we did not point out that Braxton's product was not designed for use in a research environment. It is likely that some, or many, of technical challenges we encountered can be attributed to our attempt to use that simulation in an application for which it was never intended.

Ultimately, these hurdles put progress towards successful development of the HSI testbed at risk. A decision was made to forgo inclusion of the Braxton simulation in order to focus efforts and resources on the primary goals of this program: the design and implementation of HSI features and the development of tools needed to successfully conduct research. Unfortunately, several months of effort were spent attempting to integrate the COTS satellite simulation.

Once the determination was made to end efforts to include this satellite simulation, a number of alternatives were explored. These included the possibility of using a COTS satellite simulation from other vendors and possible re-use of a Government-owned simulation. The COTS simulations were rejected because they had some of the same technical and cost drawbacks as did the original model. Specifically, we could not readily gain access to the code at the level we needed to collect data or to control anomalies. At that time, no suitable Government-owned simulation was identified. (We have subsequently learned that NASA Goddard's REACH space control simulation has a satellite simulation. We have not evaluated this simulation to determine if it could be used to upgrade the testbed.)

In order to refocus the program on the HSI issues, and to get back on track in terms of schedule MTI developed a "thin simulation" to replace of a COTS satellite simulation. This thin simulation generates data values for all of the variables in the Link 1 communications, Link 2 communications, electrical, and propulsion subsystems in a Defense Satellite Communications System (DSCS) satellite (Figure 4). Both continuous and discrete variables are simulated.

CERES personnel provided MTI the set of variables, along with the nominal, warning, and caution limits for the continuous variables used in this program. Appendix 3 contains tables listing the variables that are simulated. Where applicable, the nominal, warning, and caution ranges are indicated.

DSCS was chosen for use in the testbed as its systems are representative of many of the satellites currently flown by the Air Force. Additionally, CERES is flying a "retired" DSCS satellite. This provides a pool of experienced DSCS controllers who could participate in future research.

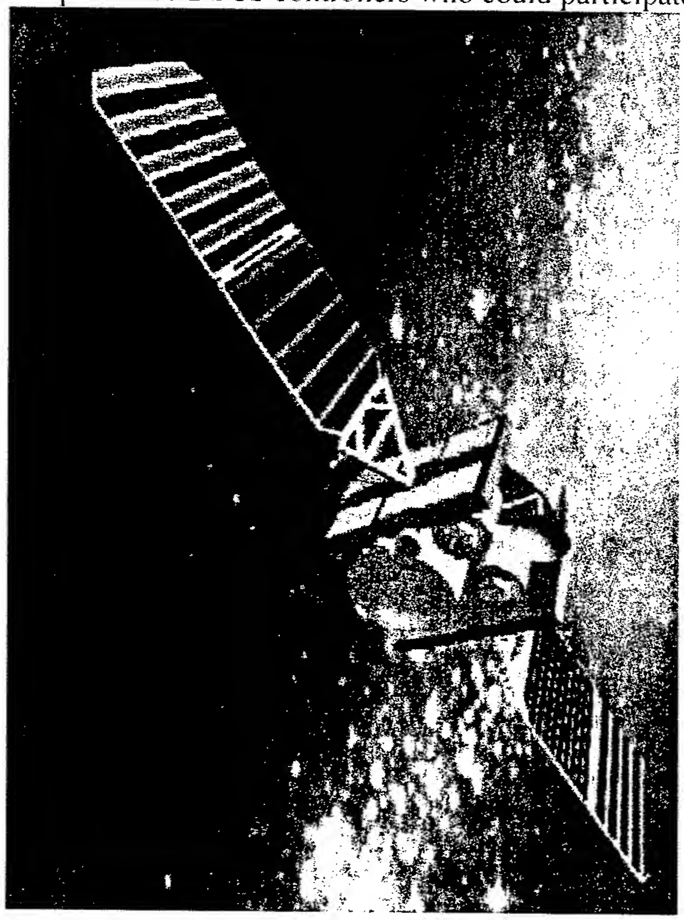


Figure 4. Defense Satellite Communications System (DSCS) Satellite.

Simulation of the continuous variables is done using a "sum of sines" technique. This technique allows the value of each variable to vary independently over time. The changes appear to the controller as being essentially random about a mean value. By changing the frequency, phase, and/or amplitude parameters of any of the sine waves used to generate a variable's value, the experimenter can change the manner in which that variable changes.

Simulation of the discrete variables is accomplished simply by assigning a default value of zero or one. These default values are contained in an ASCII text "include" file on the "Control Point" machine. The path and name of this file is "d:\PostBTI\TimeClient\DiscreteDefaultList.h. The experimenter can edit this file to create the desired initial conditions for a particular trial using any text editor.

One benefit of this approach is that the "thin simulation" is not computationally intensive. This allowed us to run the testbed on only two machines, rather than three as would be required to use a high fidelity simulation. By running on two machines the inter-machine communications requirements were reduced and simplified.

Another benefit of this approach is that unlike using a COTS satellite simulation, the source code for the simulation is available, and can be changed to meet future requirements. This makes it relatively easy to implement new HSI features, and does not require contracting with any outside source to make those changes. This capability reduces the amount of time needed and the costs of integrating new equipment.

Touch Screen Interface. The initial physical layout of the satellite control testbed was modeled after the multi-modal workstation developed by the Navy for command and control. The Navy work had showed that in their application, a large, touch screen equipped display area served the needs of the uses better than a smaller area. The Navy configuration consisted of two display surfaces located side by side above a single display area centered below the other monitors. Similarities between the command and control tasks and the tasks performed by satellite controllers, we began development of the testbed using a configuration that is physically similar to the Navy's. This allowed us to order some long lead-time hardware early on, thus avoiding delays.

As it turned out, a number of technical challenges emerged when integrating the hardware. One of the biggest challenges turned out to be due to the availability of a touch screen driver that would allow this screen configuration in the Windows 2000 operating system environment. The options available to overcome this problem were:
(a) obtain a Windows 2000 compatible driver from the vendor
(b) write a driver in-house
(c) switch operating systems

The touch screen vendor (ELO Systems, Fremont, CA) was contacted and we were advised that a driver did not exist at that time. However, they claimed to be working on the development of a driver and we were put on a list to receive a beta version when it became available. To date this driver has not been released for testing or use.

Option (b) was considered, but rejected because it posed too much technical risk. Our concern was that if the vendor had not been able to provide a driver for this popular operating system, then there were likely to be one or more "show stoppers" that we would also encounter. This concern lead us to conclude that writing a driver in-house would consume considerable amount of time without a high probability of being successful. Therefore, we decided not to pursue this option.

We elected to go with option (c) and switched to Widows 98 as the operating system for the "Control Point" machine. This meant that the voice I/O system had to operate under the older operating system. It also meant that we would now have a system of heterogeneous operating

systems. Both these consequences caused complications for the development of the testbed software.

Voice I/O. The voice I/O system used in the testbed is *Dragon Naturally Speaking*™ (Lernout & Hauspie, Burlington, MA). During the course of this program the parent company, which is based in Belgium, apparently encountered legal and financial difficulties. Partially as a result of those difficulties, the vendor's technical support for this product was often poor or non-existent. In the absence of reliable technical support from the vendor, we were often forced to rely on user's groups to obtain suggestions on how to overcome the difficulties we encountered. A more reliable source of technical information would have allowed us to avoid many of the dead ends we went down with the voice I/O software development. (NOTE: In November 2001 the Speech Division of Lernout & Hauspie was acquired by Scansoft of Peabody, MA)

A great deal of time and effort was required to get the accuracy of the voice recognition system up to an acceptable level of accuracy. In part, this was due to the vocabulary used in the control of satellites. More specifically, the names of the variables were often quite similar (e.g., "CT1AMT" and "CT1BMT") and were easily confused. The context of inquiry for the state or value of a variable did not aid the recognition engine in distinguishing between these names.

Three approaches were used to improve the speech recognition accuracy. First, the speaker dependent voice training was performed. This training consists of using the training capabilities built into *Dragon Naturally Speaking*. This requires the speaker to create a user profile and to read selected text aloud.

The second approach was to create a custom dictionary of terms. Here, we entered all of the variable names and their phonetic spellings into *Dragon Naturally Speaking*. For example, for the phonetic spelling of "CT1BMT" we use "sea tea one bee em tea." We then trained the voice recognition system by dictating the variable names and correcting the system each time an error was made. This process required many hours.

The third approach we used to improve accuracy was to include the phonetic spelling of each phrase to be recognized in the code. (These phrases are in d:\postBTI\TimeClient\MessageMap.cpp on the Control Point machine.) Below are several examples.

- *"sea tea one bee em tea"*
- *"Show me the value of sea tea one bee em tea"*
- *"Tell me the value of sea tea one bee em tea"*
- *"What is the value of sea tea one bee em tea"*

This constrains the system to recognizing only those phrases contained in this file. Other phrases are not recognized by the voice system. (There are some exceptions. *Dragon Naturally Speaking* has a number of phrases built in that are recognized regardless of any other applications that are running. For example, the phrase "*Open Word*" opens the Microsoft Word program. To date, we have been unable to identify a method for turning off these built in phrases so that only the vocabulary in this application is recognized.)

As can be inferred from the above discussion, *Dragon Naturally Speaking* does not perform true "natural language recognition". The speech engine doesn't understand the meaning of the words spoken by the controller. In order to make the system act as if it was performing "natural language recognition" we created a vocabulary list with numerous phrases that have the same effect. This allows the controller to use phraseology that is more conversational. For example, the controller could use any one of the following statements to obtain a display of the value of variable PHT-HT.

"Show me pea h tea minus h tea"
"Show me the temperature of the minus ex hydrazine line high level thruster"
"Show me the minus ex hydrazine line high level thruster temperature"

Simply replicating an existing statement and replacing the phrase to be recognized with the new one can extend the number of synonymous statements. This can be done with any ASCII text editor.

The system can respond to a controller's query regarding the value of a continuous variable in three ways:
(a) Announce the current value using the voice synthesis system
(b) Display a gauge showing the value of the variable
(c) Announce the value and display a gauge.

In order to make it easy for the controller to select which response the system will employ, the following convention was implemented:
- If the controller says just the variable name or a phrase that begins with "Tell me ..." then the system will announce the current value of that variable.
- If the controller says a phrase that begins with "Show me ..." then the system will display a gauge showing the value of the variable.
- If the controller says a phrase that begins with "What is ..." then the system will both announce the value of the variable and display a gauge.

The controller can also use voice commands to change the state of discrete variables. The discrete variables commonly represent satellite controls that the controller

Unfortunately, there are two unresolved software anomalies in *Naturally Speaking* that limit its usefulness in this application, as well as in other mission-critical systems. The first anomaly is that *Naturally Speaking* monopolizes the CPU when it is performing recognition and synthesis. This means that processing for other applications running concurrently on the same CPU is halted until *Naturally Speaking* has completed its task. Here, this means controller inputs are not processed as rapidly as they should. It also means that actions involving communications between systems in the testbed are deferred until the voice I/O system has completed its processing. This impacts the data collection system, where several frames of data may be lost while voice I/O processing is being performed. It impacts the HSI because the controller is unable to interact with the system until the voice system relinquishes control of the CPU.

A second problem is the reliability of the voice I/O system. In the testbed it is not uncommon for the voice I/O system to "hang" in a state that eliminates communication between the application and the voice system. When this occurs, it appears to the controller as if the recognition system is still functioning, but that the synthesis has failed. We have determined that *Naturally Speaking* has failed internally and in no longer returning to the application.

Further investigation leads us to believe that this problem is due, at least in part, to a problem in synthesizing integer and floating point values. When the synthesis engine is asked to pronounce a value such as "5.3 volts" the system often fails. However, if the system is asked to pronounce the same value but it is expressed without numerics, as in "five point three volts", then the system operates normally. This problem does not occur when voice synthesis of floating point values is not required. "Show me..." commands are in this category as are commands that change the state of variables having discrete values. Additionally, commands to bring up and remove screens are not affected by this problem in the voice synthesis system. We have brought the problem of synthesizing floating point values to the attention of Dragon's technical support group, but so far have not found any resolution.

All in all, it appears that voice I/O is a powerful part of the HSI for satellite controllers. However, the current implementation, while it uses a state of the art COTS system, has limitations that adversely impact its utility in this setting. Until these problems are resolved, we consider voice synthesis to be insufficiently robust for use in a mission critical system.

SYSTEM MANUAL

A System Manual for the testbed is contained in Appendix 4. This manual describes the architecture of the system and all of the major software modules.

SOFTWARE USER'S MANUAL.

Appendix 5 contains a Software User's Manual. In this manual are descriptions of the processes to be performed in order to create or modify:

- Voice commands
- Pass Plans
- System Diagrams
- Data Displays (e.g., graphical displays of a variable's value)

It is expected that this manual will be used by persons familiar with programming in C or C++ in a Windows environment.

Many of the tasks described in this manual make use of the *Design* software package (Altia, Colorado Springs, CO). It is highly recommended that the tutorials in the *Design* package be completed prior to developing new displays for the testbed.

Design of Experimenter Workstation

In the design of the testbed the needs of the experimenter have been considered. The key functions available to the experimenter are:

1. Control the start, stop, and pausing of the simulation
2. Name the data files
3. Select the data collection rate appropriate for the specific study
4. Control anomalies and malfunctions
5. Add markers to the data stream to allow easy identification of significant events
6. Add annotations to the data files
7. Select the variables to be collected
8. Set the initial conditions for the simulation

Items 1 through 6 are performed using an experimenter workstation. Items 7 and 8 are performed off line.

<u>EXPERIMENTER WORKSTATION.</u>

The experimenter interacts with the workstation with a keyboard and either a mouse or touch screen. Appendix 6 contains an Experimenter User's Manual for the workstation.

Figure 5 shows the appearance of the experimenter workstation display. Each field in this display is described in Appendix 6. Additionally, the processes of activating and deactivating anomalies are described in a step-by-step fashion in that manual.

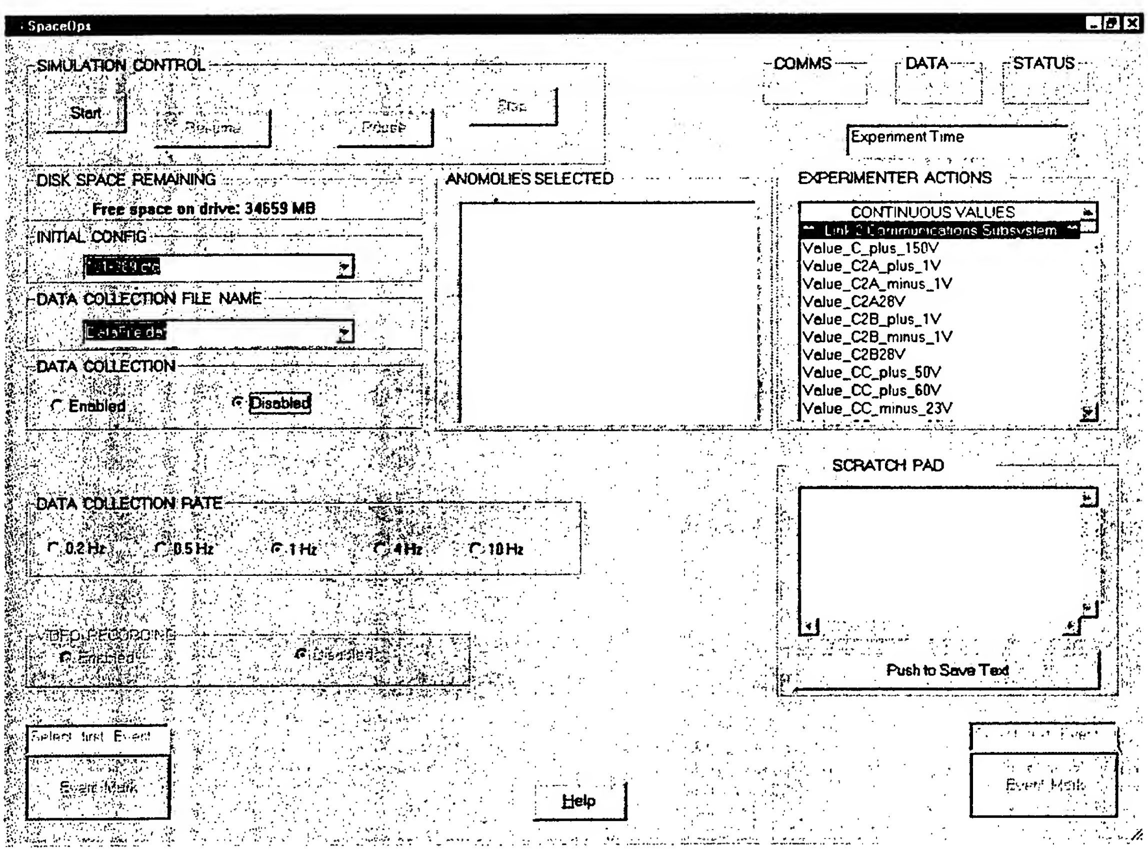


Figure 5. Experimenter's Workstation.

This figure shows the display prior to commencement of an experiment. There are no anomalies are active and the experimenter has not yet entered any event markers or annotations

DEFAULT STATE LIST

The default state of the simulated satellite can be configured as required by the experimenter. The default state of each of the discrete variables is set to either 0 (zero) or 1 (one) at the start of each simulation event. The meaning of zero or one depends on the particular variable. (For example, a value of 0 could mean "off", "encoded", or "20 watts", depending on the particular variable.) The default values for all of the discrete variables in the simulation are contained in *d:\PostBT\TimeClient\DiscreteDefaultList.h.* This is an ASCII text file that can be edited by the experimenter. In order to change the default state of a specific variable the experimenter can simply change the value from 0 to 1 (or from 1 to 0). Alternatively, the controller may "comment out" the assignment and replace it with a line containing the new value. When "commenting out" a line in this text file the C-language convention of using a double slash (i.e., "//") before any text is recommended. Changes in this file do not have an effect until the simulation is recompiled.

DATA COLLECTION LIST

The experimenter can specify the data that will be collected during a simulation. The names of the variables that are being collected are contained in *d:\PostBT\TimeClient\DataCollectionList.cpp.* If the name of the variable does not appear in this file, or if the line containing that variable name is "commented out" then that item will not be collected. It is recommended that the experimenter "comment out" variables when they are not desired, rather than deleting them from this file. It is recommended that the C-language convention of using double slashes (i.e., "//") be used for commenting out text.

This is an ASCII text file and can be edited by the experimenter. If the simulation of the satellite is enhanced so that new data are available, the experimenter needs to add those variable names to this list. This is simply a matter of copying an existing entry and editing the name of the variable.

The order of the variables in the data file will be the same as the order of the variables in this list. For example, the n^{th} variable in the list will be the n^{th} column of the data file.

HSI Technology Review and Projection

SPEECH AND MULTIMODAL INTERFACE TECHNOLOGIES

Developments related to advancements in HSI technologies have been closely monitored and documented in reports and project review meetings throughout the duration of this program. An

initial technology assessment was performed during Phase I to identify a reasonable set of emerging HSI technologies that should be considered for inclusion in the SOC testbed during the Phase II timeframe. The selected candidate technologies included several well-researched advanced HSI technologies that were beginning to transition from the laboratory testing phase into trials in a variety of operational environments. Based upon discussions with the program sponsor and members of the user community it was agreed that the technologies incorporated in the current SOC testbed represent a reasonable set of advanced HSI technologies commensurate with the Phase II budget and schedule constraints. A detailed review of the HSI technologies selected for the Phase II SOC testbed was conducted prior to implementation to provide a better understanding of the conditions under which the selected technologies applied individually or in certain combinations, tend to enhance human-system interaction and task performance. The review also examined the conditions under which the selected technologies have failed to improve operator performance, as well as potential implementation pitfalls. The results of this review were combined with the results of the task and user requirements analysis activity to guide decisions regarding SOC testbed architecture and implementation.

The current SOC testbed provides an excellent foundation for the further development and evaluation of spoken language interfaces, one of the more promising avenues for achieving a major breakthrough in the communications between humans and computer-based systems. The emerging spoken language interfaces will allow users to communicate with computer-based applications through words and phases, our most familiar form of everyday communication. Spoken language interfaces have the potential for greatly reducing the skill level and training requirements for satellite controllers, and significantly increasing their mission performance. They also have the potential for enhancing usability and situation awareness by allowing many operations to be performed without hand or eye movements that distract from the task-critical information being presented via visual displays. The following section provides a brief update on the current state-of-the-art and trends for speech technology, as well as recommendations aimed at providing a roadmap for near-term (the next two or three years) SOC testbed development.

State-Of-The-Art

Since the current program was initiated there has been a significant increase in speech product capabilities, the introduction of new products incorporating speech technologies, and unprecedented investment in speech technology research and development. Currently the most widespread applications of speech technology are the use of continuous speech computer desktop products for dictation, and the use of automated speech recognition (ASR) systems for customer service and self-service applications (e.g., travel reservations, telephone directory, stock trading, and banking). The existing touch-tone interactive voice response (IVR) systems that have frustrated users with their tediously long menu-based navigation via the telephone keypad are rapidly being replaced by telephone-to-computer speech recognition systems. These ASR systems (e.g., SpeechWorks, Nuance, and Entropic) incorporate the latest robust speaker-independent continuous speech recognition technologies and sufficient dialog capabilities to handle various routine tasks normally performed by humans, including the funding of numerous start-up ventures.

In addition, each of the major consumer speech products companies (i.e., IBM, Dragon Systems, Phillips, and Lernout & Hauspie) has introduced second-generation large vocabulary (i.e., in the range of 250,000 words) continuous speech recognition dictation products. Unlike the ASR speaker-independent systems, these speech systems require each new user to perform at least one "enrollment" or training session that can vary from 10 to 30 minutes. These training sessions are to train the speech system and not the user. The recognition accuracy for these speaker-dependent systems continues to improve over time with normal use and additional training sessions. Dictation places great demands on the speech recognition technology, driving the suppliers to introduce significant improvements in user interface features and recognition accurately in each new product release.

The current SOC testbed incorporates a limited, but potentially useful subset of the spoken language technologies, including speech recognition and speech synthesis. Speech recognition is being used primarily to provide controller with direct access to information via simple English speech commands. This capability should offer a performance advantage over traditional Graphical User Interface (GUI) point-and-click selection from an ever-growing array of menu options, buttons and icons as employed by current spacecraft control systems. As currently implemented for the SOC testbed, speech synthesis technology is mainly used to provide an alternative means of providing controller-requested information regarding the current value of a particular system variable. This allows controllers to get access to important information while attending to visually-oriented tasks.

The speech recognition technology employed in the current SOC testbed represents the state-of-the-art with respect to available speaker-dependent commercial-off-the-shelf (COTS) products. The Dragon Systems *Naturally Speaking* software that supports all of the SOC's speech features is the product of more than 10 years of research and development supported by several million dollars of DoD and industry funding. The latest release of *Naturally Speaking* provides many speech input features and high-level tools for developing and optimizing the performance of speech-enabled applications running on the Microsoft Windows operating system. Typical of current leading-edge speech products, Naturally Speaking has its share of undocumented technical problems. However, we have been able to achieve very high recognition accuracy for an application with a relative large vocabulary with a high potential for speech recognition errors. Despite a period of business turmoil, the Dragon Systems R&D group was able to make continuous improvements to the *Naturally Speaking* product since it was initially selected as the speech technology for the SOC testbed. The *Naturally Speaking* technology is now owned by a stable company, ScanSoft Inc., and there is reason to believe that it will remain a state-of-the-art speech technology product in the foreseeable future. The array microphone used in the current SOC testbed is one of better low-cost microphones of its kind in terms of speech recognition accuracy.

Near-Term Development Recommendations

The task analysis reported in Appendix 1 revealed several specific shortcomings with current SOC user interfaces, and specific recommendations regarding the potential uses of current speech technology. In addition, the results from recent successful applications of speech interface technology for military control center environments (e.g., the USAF TapTalk, the

SPAWAR Multimodal Watchstation, and the DARPA CPoF programs) suggest the general types of tasks that are likely to benefit from the application of current speech technology.

Generally Increase Speech-Enabled Tasks. Based upon the above findings, it is recommended that the SOC testbed should be used to explore the potential benefits of additional speech input & control for satellite control tasks involving direct access to frequently performed functions (navigation), entering satellite parameters for labeled fields (data entry), and accessing telemetry history data to resolve anomalies (database query). Based upon the combined findings from the task analysis and the key HSI research and development programs, it is recommended that the near-term plans for enhancing the speech interface capabilities focus on making use of the speech processing capabilities supported by the current SOC testbed to incrementally expand the application of speech recognition and synthesis to (1) perform tasks that are inherently awkward to perform with a keyboard or mouse, (2) to provide a more balanced workload across the sensory modalities, and (3) to reduce the demands on the operator's memory by eliminating multiple menu selections and providing direct access to needed information.

Implement SuperMOCA Control Interface Language Specification. A specific example of the recommended approach is illustrated in a very recent report by Remington and Coven (2002). The authors describe a specific task performed by many spacecraft controllers and spacecraft test engineers that could benefit from the implementation of current speech technology. This report points out that the creation of directive macros with one of the cryptic commanding languages currently in use (e.g. STOL, GOAL, CSTOL, and ETOL) via a keyboard is a very tedious and error-prone task. Remington and Coven developed the Spacecraft Speech Command Scripting Prototype to demonstrate how speech recognition could be used to prepare directive and command macros, taking advantage of speech feedback for guidance, verification and error correction based upon the Control Interface Language (CIL) Specification.

Developed under NASA's SuperMOCA project, CIL provides a comprehensive specification for a next-generation language with an English-like syntax, an object-oriented database architecture, and natural language command parsers for use in future spacecraft test and operations systems. The language can be used to control all of the activities associated with the space vehicle and their supporting ground systems. This includes monitoring and control at the low-level of individual actuators and sensors (e.g., "open valve 1") and at a high level suitable for systems that have intelligence to carry out complex actions (e.g., "reorient spacecraft to nadir pointing").

The Spacecraft Speech Command Scripting Prototype also demonstrates several other uses of speech technology in the context of the task of creating spacecraft commanding macro scripts including the application of continuous speech recognition to support the ad hoc dictation to attach speech annotations to macros and the use of speech output to provide attention-getting feedback and guidance. Like the SOC testbed, the Spacecraft Speech Command Scripting Prototype implementation incorporates the *Naturally Speaking* product and the ActiveX components provided by *Naturally Speaking* Software Developer's Kit (SDK) to create the speech user interface and to perform the fundamental speech recognition and text-to-speech synthesis processes. Implementation of NASA's no-cost CIL specification and parsers would represent a strategic step forward for the SOC testbed. It would set the stage for the future

development of a true natural language interface for the SOC, one that goes beyond word-spotting speech recognition to one that is capable of limited understanding of spoken language.

It is worth noting that there is some debate regarding the value of natural language recognition systems in mission critical applications. Just as in systems where humans communicate with one another, it may be beneficial to standardize and constrain the vocabulary. A constrained vocabulary, like that implemented here, reduces the likelihood that a command or communication will be misinterpreted. It also encourages brevity. Both these attributes are of value in the context of a satellite control system in which communications are between the human and the machine.

Provide In-Context Speech Command Guidance. The Spacecraft Speech Command Scripting Prototype incorporates additional speech interface features that should be given consideration as potential near-term enhancements to the SOC testbed implementation. For example, it addresses one of the major shortcomings of advanced speech user interfaces. Unless speech is merely used as an alternative means of activating on screen controls, the available speech commands are not typically visible, and therefore users do not necessarily know what speech commands are available. This problem can be addressed with the introduction of the "What can I Say?" feature. At any time during the performance of an application a user can say, "*What can I say*" to bring up a list of the spoken commands that are available at that point in the application. In some respects this is an improvement over the typical GUI approach in which a user often has to explore hidden second and third level dropdown menu options to determine which functions are currently active (i.e., looking for options that are not "grayed-out"). An in-context "What can I Say?" feature can be used as a learning-aid, reducing the need to access on-line help and reference documents.

Take Advantage Of New Microphone Technologies. In addition to speech capabilities, it is also recommended that future plans for improving the SOC testbed include a policy to routinely take advantage of the relatively high return on investment normally associated with the steady improvements in microphone technology. This is of particular importance for the ultimate success of the SOC program. Overcoming the potential difficulties of achieving acceptable speech recognition in a relatively high-noise space control center operational environment is dependent upon improvements in both noise canceling algorithms and microphones. Fortunately, advancements in noise canceling and directional array microphone technologies are keeping pace with speech processing improvements. For example, in the past month Acoustic Magic introduced their *Voice Tracker* microphone with variable scanning range settings.

In the narrow-angle mode, the array microphone will scan +/- 45 degrees which practically eliminates the impact of talkers and noise sources outside this region. In addition, the *Voice Tracker* can be set to turn off during periods of silence and when the sound source is located outside the array's narrow field of "view." The array turns on again very quickly when the user resumes talking. In other words, these new features can be used to reduce the negative impact that spurious words or sounds from sources outside the listening area normally have on speech recognition. The improvements in speech recognition accuracy typically achieved from microphone technology enhancements are well worth the modest cost of most new microphones. For example, the new *Voice Tracker* can be purchased for about $200. In addition, use of the

SOC testbed for experimentation with various microphones, including those developed for use in other AFRL programs involving speech recognition would be a near-term project with potential benefits to multiple programs.

Provide Limited Natural Language Processing Capability. The current SOC testbed speech recognition technology makes use word-spotting techniques that require the multi-word commands to be spoken in a fixed order. With the addition of a natural language processing capability the user could speak naturally, using whatever form of words that come to mind. This removes the mental workload associated with having to remember the exact order in which the words must be spoken. Commercial tools for building future spoken language interfaces incorporating a complimentary set of natural language processing and speech recognition technologies are now becoming available. The use of such tools should enable the SOC to recognize and understand the meaning of human speech within the context of application domain, perform the appropriate action, and to carry on a relatively natural mixed-initiative dialogue with the user. We will not come close to achieving the level of language understanding and natural dialogue exhibited by HAL in the movie *2001 A Space Odyssey* in the foreseeable future. But the next-generation SOC should provide operators with direct access to functionality, information, and computer-assisted decision aids to rapidly and accurately perform critical operations such as re-tasking and complex maneuvers. Like HAL, the SOC could make use of currently available speaker identification and verification technology to allow the system to recognize who is speaking to it and to control user access.

MULTI-MODAL USER INTERFACES

The most noticeable trend in the advancement of human-systems interfaces is the combination of individual HSI technologies to form multimodal user interfaces. Speech technologies represent the cornerstone in many of the new multimodal interface implementations. Multimodal user interfaces represents a shift away from conventional GUI interfaces toward providing users with greater expressive power, naturalness, flexibility, and portability. The findings from several studies indicate that well-designed multimodal systems can integrate complementary human sensory modalities to yield a highly synergistic blend in which the strengths of each mode are capitalized upon and used to overcome weaknesses in the other. There is mounting evidence that systems which integrate multiple user interface technologies such as natural language speech recognition, speech output, gesture recognition, touch input, haptic feedback, eye & head tracking, and spatial audio have the potential to significantly improve user performance for a variety of tasks. The performance enhancement potential of multimodal interfaces is mainly attributed to their naturalness and the increased bandwidth with the use of multiple sensory channels. Increased bandwidth simply means communicating more information per unit of time, other things being equal, improves the efficiency of human-system interaction and increased user performance. While it seems likely that the SOC of the future will have a multimodal interface, it is not clear what combination of advanced HSI technologies will be required to meet future space mission performance objectives.

The SPAWAR Multi-Modal Watch Station (MMWS), described in a previous HSI technology review report, still represents the state-of-the-art with respect to a multimodal implementation of a SOC-like command and control console. Briefly, the MMWS is comprised of multi-modal control and input methods including touch, speech, and eye-tracking combined with multiple

flat-panel displays, 3D audio, and advanced information management technologies. The MMWS testbed was used to conduct a series of experiments to examine various control and display configurations including pull-down menus, off-screen function keys, on-screen function keys, voice and trackball. Based upon operator speed and accuracy and preference data it was found that the design options using touch screen, voice entry, and touch entry function arrays were among the fastest methods. Pull down menus, as found on most commercial software products, were among the slowest function activation methods tested. Combinations of voice and touch activation with function key activation by alternate hand was found to be effective in distributing workloads. Most of the current advanced multimodal user interface projects focus on mobile applications involving wearable computers, or command post applications that involve extensive interaction with maps. An example of the latter is Oregon Graduate Institute's RASA multimodal command post project that fuses spoken and gestural information to interact with maps. For example, RASA users can draw a line on the map and speak or draw its type "FORTIFICATION," draw a closed curve and speak "LANDING ZONE ZULU," or point at the map or entities on the map and ask questions such as "WHERE IS SCOUT SIX?"

Influenced by the SPAWAR Multi-Modal Watch Station project, the current SOC testbed architecture was designed to accommodate a multimodal interface implementation consisting of several input and output devices. As an example, the current SOC testbed includes a touch screen interface as an alternative to the traditional mouse interface. As described earlier, this interface allows the controller to simply touch the desired location on the screen and the effect will be the same as performing a mouse click at that same location. This basic implementation of a touch interface, when combined with the available speech interface capability sets the stage for future multimodal user interface development and experimentation. Multimodal user interfaces present many options in terms individual HSI technologies and various combinations. Some of the leading candidates with respect to their maturity, potential applicability, potential benefits, cost, and development effort are presented below.

Speech-Touch Interface

The development of an integrated speech-touch capability would appear to be the next logical multimodal enhancement. With this capability the operator could make use of the touch sensitive screen and voice input to access information regarding a displayed system component by merely touching the graphical object representing the system component and issuing a verbal command or asking a question. For example, the operator might touch a graphical representation of a communication link and ask, "How much bandwidth is available?" Together, the speech and touch modes have been shown to provide complementary capabilities permitting users to engage in more powerfully expressive and transparent information-seeking dialogues to provide flexible descriptions of objects, events, spatial layouts, and their interrelation. Compared with speech-only interaction, empirical work with users during visual-spatial tasks has demonstrated that multimodal touch/speech interaction can result in 10% faster task completion time, 36% fewer task-critical content errors, 50% fewer spontaneous disfluencies, and also shorter and more simplified linguistic constructions with fewer locative descriptions (Oviatt, 1997; Oviatt & Kuhn, 1998).. The addition of an integrated touch-speech capability would probably improve SOC operator performance for a variety of tasks with a relatively small development effort.

Speech-Gesture Interface
A natural extension of the speech-touch capability would be the addition of gesture recognition product. The major development effort would be associated with integrating the gesture and speech capabilities. Multimodal user interfaces with combined human speech and gesture modalities have been subjected to far more research than any other combination of the advanced HSI technologies. These applications range from military map-based and virtual reality systems for engaging in simulation and training, to field medic systems for mobile use in noisy environments, to standard text-editing applications. A study comparing the gesture/speech system with a standard GUI reported a 3.5-fold speed improvement in the average entity creation time, including overall error handling. The mean time to repair error was 4.3 times faster when interacting with the gesture-speech interface (Cohen, Johnston, McGee, Oviatt, Clow, & Smith, 1998). These findings regarding the improved interaction with maps and simulation setup have possible implications for the space operations domain. Several planning and analysis applications used by various space system operators and analysts involve world maps and/or simulation components (e.g., STK and ComPlan). A task analysis of these and related space applications would likely reveal space operations tasks that could potentially benefit from the application of gesture-speech interface technology.

Spatial Audio Feedback
The SOC testbed could be used to gain experience with spatial audio technology. Spatial audio could be used to direct an operator's attention to a particular screen in the current duel screen display implementation. A possibly even better application for spatial audio is intelligibility when multiple sources must be monitored. It is much easier to understand and identify a person when they do not share the same position in the audio field Nelson, (Bolia, Ericson, and McKinley, 1998). The cost of spatial audio is very small since it has recently become a consumer item used to enhance the realism of computer games. The use of synthesized speech as an audio source is another capability that is inexpensive to implement. Most of the commercial speech recognition packages provide an integrated text-to-speech feature for no additional cost. It is likely that the SOC testbed will be used to explore 3D spatially-oriented technologies such as volumetric displays, and partial immersive virtual environments. In this case the use of the spatial audio technologies will become more attractive. However, the most compelling reason for early implementation of spatial audio is that it represents a low-cost low-risk capability that will undoubtedly generate some out-the-box thinking regarding the ways in which 3D display concepts might be applicable to the space operations domain.

Eye-Tracking Interaction.
Incorporation of an eye-tracking capability within the SOC testbed would provide a potential useful alternative form of interaction. Eye-movement and eye gaze interaction should offer an easy, natural, and fast way of interacting with the multiple large displays used in the current testbed. In addition, eye-tracking provides a very direct means of pointing to an object which can then be manipulated or further explored with the expressive power of speech. There are several relatively accurate and unobtrusive commercially available eye-trackers that use a camera and an IR light source to track gaze by computing the angle between the corneal reflection and centroid of the pupil. Driven by potentially lucrative automotive applications, the technology is steadily improving and the relative high cost of the current COTS products is expected to rapidly decrease.

The recommended SOC testbed enhancements outlined above would provide a relatively advanced multimodal user interface testbed comprised of a complementary set of emerging HSI technologies. These specific HSI technologies were selected on the basis of their perceived applicability, maturity, availability, costs, and sensory load balancing potential. The resulting next-generation SOC testbed should provide a valuable vehicle for exploring emerging multi-modal interfaces that allow information exchange in ways familiar and comfortable to the user, principally through natural actions in the sensory dimensions of sight, sound, and touch.

Summaries of the HSI research and development efforts related to the HSI technologies recommended for near-term adoption, as well as many other emerging HSI technologies are presented in Appendix 7. The major well-funded HSI research and development programs that are likely to yield integrated operational solutions are described in more depth than the less ambitious projects that tend to focus on an individual technology. In any case, the information provided in Appendix 7 should provide a useful point of departure for future HSI technology monitoring and assessment efforts.

The absence of other promising advanced HSI-enabling technologies from the near-term roadmap presented above does not preclude their implementation as part of the next-generation SOC testbed. The testbed architecture has been designed with a view to incorporate a wide range of emerging alternative control and display technologies, as well as software-based operator aids and performance measurement tools. The architecture provides the flexibility to accommodate other HSI technologies that might be highlighted as a result (1) technological breakthroughs, (2) additional task and mission needs analyses, (3) new on-board spacecraft capabilities, (4) increased automation of current manual control tasks, and (5) a reduction in the current uncertainty regarding future space systems operational concepts.

Evaluation of Advanced HSI

A case study was conducted using the advanced HSI developed during this program. A research note describing the evaluation in detail is contained in Appendix 8. Highlights of that report are contained here.

OBJECTIVES

The main objective of this evaluation was to obtain quantitative measures of the performance changes attributable to the advanced, multi-modal HSI.

A secondary objective of this effort was to demonstrate the use of the testbed in a HSI research tool.

HSI DESCRIPTION

The baseline condition was based on the COBRA interface used by CERES. The baseline interface used in this study was enhanced over the version used in CERES. Some of the major enhancements were:

- Reduced number of mouse clicks required to send a command

- Single mouse and keyboard to interact with the system (vice two required in the operational COBRA interface)
- Embedding data values directly in the pass plan
- Presence of a continuous monitor alerting the controller to variables outside the nominal range

Voice control to bring up pass plans and open telemetry display windows
The advanced HSI was described in detail earlier in this report. The advanced HSI included all of the features in the baseline HSI. It also allowed multi-modal interaction through voice recognition and synthesis and through a touch screen. The advanced HSI also included improved subsystem displays.

An experienced satellite controller participated in this evaluation. This controller performed a series of pass plans. The pass plans were developed by CERES and were intended for use in evaluations that were to have been conducted in collaboration with AFRL. A total of seven pass plans were used in this evaluation.

Following completion of all of the pass plans, subjective ratings and comments were made by the controller.

RESULTS

Objective Measures. Figure 6 shows the average time to complete a pass plan. The results indicate that the pass plans were performed faster using the advanced HSI (approximately 100 seconds per pass plan) than the baseline HSI (approximately 250 seconds per pass plan). This is a large increase in speed. When one considers the fact that in most supports several pass plans are run consecutively, it is clear that using the advanced HSI would allow the controller to perform more tasks during the time allotted for a support, or the time allotted for the support could be reduced allowing other users of the AFSCN to use those assets.

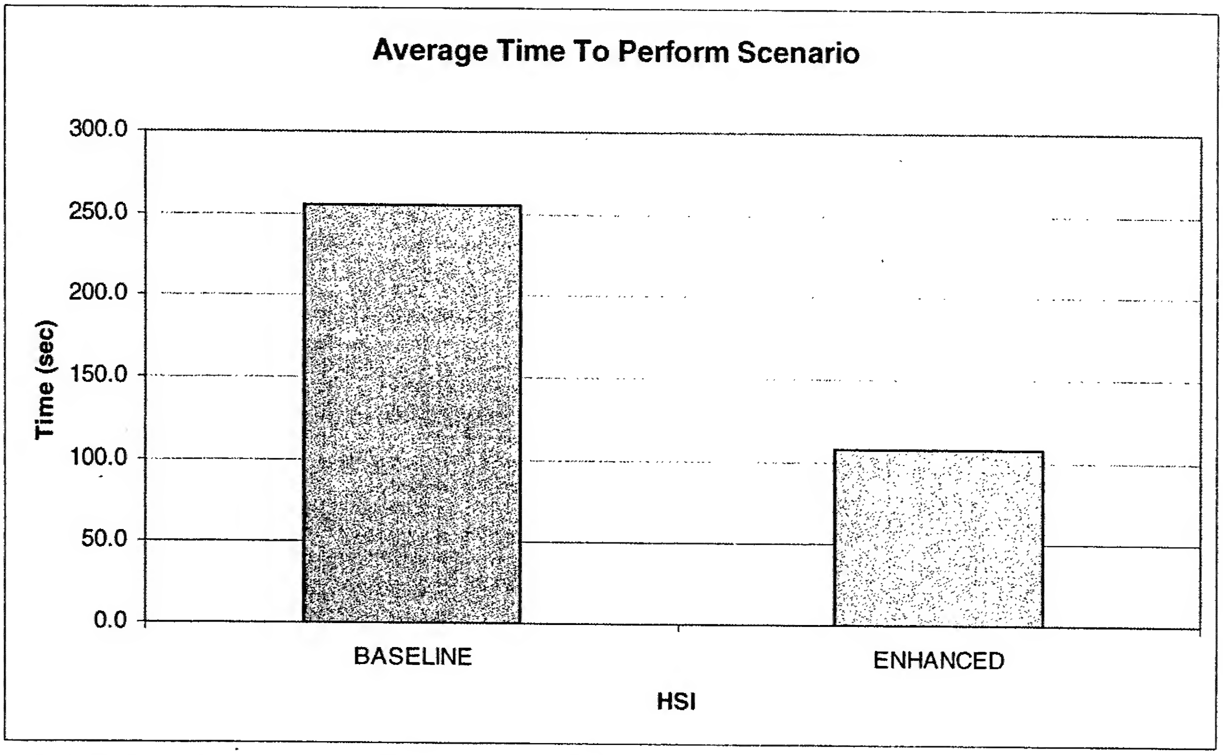


Figure 6. Average time to complete a pass plan.

The average number of mouse clicks per pass plan are shown in Figure 7. The results also showed that the controller made fewer mouse clicks in the advanced HSI (approximately 3 per pass plan) compared to the baseline condition (approximately 16 clicks per pass plan). This indicates that a voice interface though the number of mouse clicks in the baseline system used here was reduced compared to the operational COBRA system,

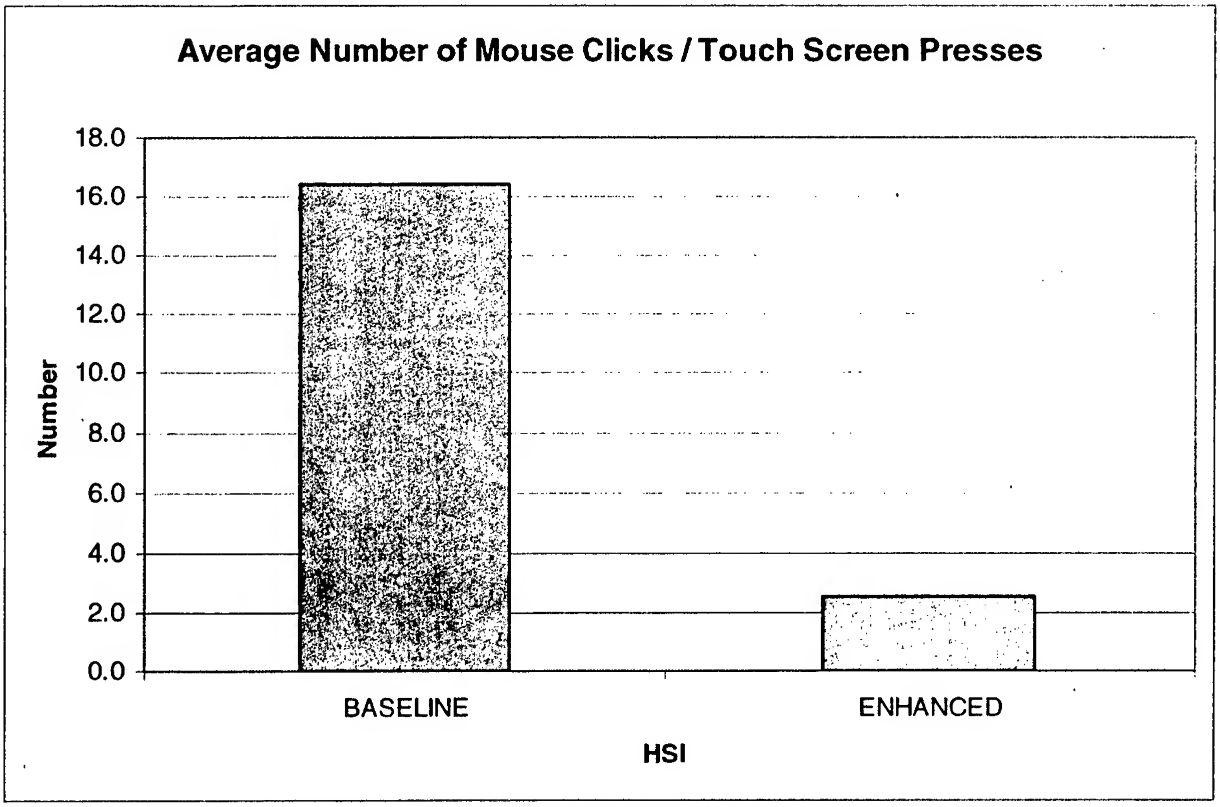


Figure 7. Average number of mouse clicks.

Examining this figure, it is clear that in the enhanced condition the controller had fewer mouse clicks than in the baseline condition. This result occurred even though the controller had the option to perform pass plans in the enhanced condition. This suggests to us that this controller had a preference for using voice commands when they were available, as opposed to selecting a process where a great deal of manual action was required. One implication of this may be that a touchscreen has only a small added value compared to a voice interface for user acceptance.

These data may artificially inflate the number of mouse clicks used in the enhanced condition. In the testbed, the voice recognition system only operates when the main window is in "focus". That is, if another window, such as a system diagram, is highlighted then the voice system will not operate. The controller must click inside the main window whenever another window gains focus. (This is an issue in the Windows environment. It is very difficult, if not impossible, to keep the focus on one specific window while others are being opened. We have not developed a solution for this problem as yet.)

Subjective Ratings. Following completion of all of the pass plan the controller made a series of subjective ratings and was then debriefed.. These ratings are describe in Appendix 8. In general, the ratings indicate that the controller liked the advanced HSI and felt that it improved his ability to perform the pass plans. Controller comments during the debriefing also indicate that the advance HSI was perceived to be an improvement. The controller did note misgivings about the ease that commands could be sent to the satellite. This probably reflects the fact that in this controller's experience the commands need to be cross-checked by others before they are sent. In part, the need for this type of cross check is reduced or eliminated by the use of pass plans and

direct manipulation of graphic representations of the satellite's controls. The reduction being due to the elimination of the need to type in a command and the attendant possibility of making an input error. (These errors can't be caused by the controller when performing a pre-approved, validated pass plan. Nor can typographical errors occur when a graphic representation of a control is manipulated. There simply isn't any typing on the part of the controller.) However, this process does not provide any check that the pass plan being performed or that the direct control manipulation is correct – the controller could make an error, albeit at a different level than a typographical error. For an operational HSI, this issue needs to be addressed.

CONCLUSIONS

This single subject study showed that the potential of an advanced multi-modal HSI to improve the performance of a satellite controller. The reductions in average time to perform a pass plan and in the number of physical actions (i.e., mouse clicks and touch screen presses) required were striking. These findings suggest that an improved HSI allow better use of satellite support time and would allow the controller to make better use of the control modalities available. However, because this study used only a single controller, these data should be considered very preliminary. These findings would need to be replicated before we could confidently predict similar performance gains in the population of satellite controllers.

Summary

The goals of this work were to design an improved HSI for satellite controllers and to develop a testbed for conducting HSI research. Both of these goals have been met.

The HSI developed during the course of this work allows multi-modal interaction. Voice recognition and synthesis and a touch screen interface are available along with a conventional mouse and keyboard interface for the controller's use. The display formats were developed to allow the controller to rapidly and accurately determine the satellite's state of health, and to determine what remedial action is required and to execute those actions. The performance with the advanced HSI showed a marked improvement over the baseline HSI. This result was found even though the baseline HSI had a number of features not in the operational COBRA interface. If the operational COBRA interface or if one of the HSIs used in the SOPs, was used as the baseline then we expect that the differences would be even more dramatic. It is expected that the combination of improvements over existing workstations will allow controllers with relatively little experience or controllers who infrequently perform supports, to perform as well as expert controllers.

The testbed demonstrated its ability as a research tool. We were able to prototype the displays and controls needed to complete the baseline and improved HSIs. Satellite anomalies were introduced into the scenarios and removed successfully. The effects of these anomalies were visible to the controller as changes in the values of the measurands displayed on the workstation. Data describing controller actions data were collected at a sufficiently fast rate to allow comparisons between conditions. Additionally, data describing the status of the satellite variables were collected allowing reconstruction of the session if needed.

Future development of the testbed should be related to the specific research needs. At this time, we can see three different development thrusts that could, depending on the nature of the HSI research being performed, be useful. The first is to increase the bounds of the simulation to include AFSCN linkages and components. The current simulation capability deals with tasks that occur once the satellite has been contacted. There are a number of pre-contact tasks that controllers perform prior to this point. For a "full-mission" simulation these pre-contact tasks should be included.

The second direction for enhancing the testbed is to increase the fidelity of the satellite simulation. As we have noted earlier, the DSCS satellite simulation is rudimentary. This level of fidelity is sufficient for examining many tasks, but there are limits to its usefulness. A higher fidelity simulation would include the interactions between the simulated variables. These interactions are now approximated by the experimenter initiating and removing anomalies during the simulation. Adding the interactions between the components would reduce the experimenter's workload and allow the simulation to more accurately show the effects of changes in the satellite's condition.

The third area where further development of the testbed is warranted is in the range of satellites simulated. Currently, only a single satellite, a generic DSCS type satellite, is simulated. Simulation of other types of satellites would allow researchers to examine HSI approaches directed at the unique features of other satellites. For example, one could imagine a new HSI approach focusing on control of satellites that use 3-axis stabilization. (The DSCS II satellites are spin-stabilized, so such a HSI would not be expected to have an effect in a simulation using only a DSCS II satellite.) Including a range of satellite simulations would also facilitate use of the testbed to explore issues that arise when controllers are tasked to fly a constellation consisting of different types of satellites. This is likely to become more common in the future than it is now, and the testbed could be used to examine the issues that will emerge.

Acknowledgements

We would like to acknowledge a few of the many individuals who contributed to the success of this program.

Mr. Chuck Jackson and Mr. Carlo Moldonado of the US Air Force's Center for Research Support (CERES) for sharing their time and expertise as satellite controllers. Without their help and assistance, the task analysis would have been impossible. These gentlemen also arranged tours of a number of SOCs and of the Pike RTS so that we would have a deeper understanding of the role the controllers play in the satellite operations system.

Major Douglas Howse for CERES for generously allowing us to spend time with the controllers and other staff at CERES.

Mr. Chad Oster of CERES shared his knowledge of satellite operations and provided us much of the information we used to create the simulated satellite control environment. His thoughts on possible solutions for important HSI problems in current satellite control workstations served as a basis for many of the screen designs in the testbed developed during this program.

Mr. John Ianni of the Air Force Research Laboratory served as the Government's Technical Monitor for this work. John's enthusiasm and support for this work were unfailing.

ACRONYM LIST

ACRONYM	DEFINITION
AFRL	Air Force Research Laboratory
AFSCN	Air Force Satellite Control Network
ASR	Automated Speech Recognition
CERES	Center For Research Support
CIL	Control Interface Language
COBRA	COTS Based Real-time Architecture
COTS	Commercial, Off-The-Shelf
DSCS	Defense Satellite Communications System
GUI	Graphical User Interface
PAP	Program Action Plan
HSI	Human- System Interface
IVR	Interactive Voice Response
MMWS	Multi-Modal Watch Station
MTI	Monterey Technologies, Inc.
RTS	Remote Tracking Station
SA	Situation Awareness
SBIR	Small Business Innovation Research
SDK	Software Developer's Kit
SOPS	Satellite Operations Squadrons

Bibliography

Bolia, R.S., Nelson, W.T., Ericson. M.A., and Simpson, B.D. (2000). A speech corpus for multitalker communications research. J. Acoust. Soc. Am. 197(2), 1065-1066.

Bolt, R.A. (1980). Put-that-there: Voice and gesture at the graphics interface. ACM Computer Graphics, 14(3), 262-270

Barbato, G. (2000). Uninhabited combat air vehicle controls and displays for suppression of enemy air defenses. CSERIAC Gateway, XI(1), 1-4.

Cohen, P.R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J.. (1997) Quickset: Multimodal interaction for distributed applications. Proceedings of the Fifth ACM International Multimedia Conference. New York:ACM Press, 31-40.

Cohen, P. R., Johnston, M., McGee, D., Oviatt, S. L., Clow, J., and Smith, I. (1998). The efficiency of multimodal interaction: A case study. In the Proceedings of the 5th International Conference on Spoken Language Processing, Sydney, Australia, 2,. 249-252.

Dockery, J.M., and Hill, J.T. (June 1997). Virtual command center. Paper presented at the Third International Symposium on C2 Research and Technology. Washington D.C.:National Defense University. (also available at http://www.dodccrp.org/Proceedings/DOCS/wcd00000/wcd0005d.htm)

Maddox, M.E. and Muto, W.H. (April 1999). Three Mile Island: The human side. Ergonomics in Design, 6-12.

McGee, D. R., Cohen, P. R., Wesson, R. M., & Horman, S. (2002). Comparing paper and tangible multimodal tools. Proceedings of the Conference on Human Factors in Computing Systems (CHI'02). New York:ACM Press, (Minneapolis, MI, Apr. 20-25).

Moray, N. (1997). Human factors in process control. In G. Salvendy (Ed.), Handbook of Human Factors and Ergonomics (pp. 1994-1971). New York: Wiley and Sons.

Nelson, W.T., Bolia, R.S., Ericson, M.A., and McKinley, R.L. (1999). Spatial audio displays for speech communications: A comparison of free field and virtual acoustic environments. Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting. 1202-1205.

Nelson, W.T., Bolia, R.S., Ericson, M.A., and McKinley, R.L. (1998). Monitoring the simultaneous presentation of multiple spatialized speech signals in the free field. In Proceedings of the 16Th International Congress on Acoustics and the 135th Meeting of the Acoustical Society of America, pp. 2341-2342.

Nelson, W. T., Bolia, R. S., Russell, C. A., Morley, R. M., & Roe, M. M. (2000). Head-slaved tracking in a see-through HMD: The effects of a secondary visual monitoring task on performance and workload". Proceedings of the IEA 2000 / HFES 2000 Congress. Vol. 3, 390-393.

Noyes, J.M., Baber, C. & Leggatt, A.P. (2000). Automatic speech recognition, noise and workload. In Proceedings of XIVth Triennial Congress of International Ergonomics Association and 44th Annual Meeting of the Human Factors and Ergonomics Society 'Ergonomics for the New Millennium' (Vol. 3, 762-765). Santa Monica, CA: HFES

Osga, G., Nugent, B. and Campbell, N. Impact of control/display configuration and graphic user interface methods on performance speed and accuracy. SPAWAR Systems Center Publication.

Oviatt, S. (August, 2000). Preferred modalities in dialog systems. Proceedings of the International Conference on Spoken Language Processing. Vol. II, 727-730.

Oviatt, S.L. (2000). Multimodal Signal Processing in Naturalistic Noisy Environments. In B. Yuan, T. Huang & X. Tang (Eds.), Proceedings of the International Conference on Spoken Language Processing (ICSLP'2000), Vol. 2, (pp. 696-699). Beijing, China: Chinese Friendship Publishers.

Oviatt, S. L. (1999). Mutual disambiguation of recognition errors in a multimodal architecture, Proceedings of Conference on Human Factors in Computing Systems: CHI '99, New York, N.Y.: ACM Press, 1999, 576-583.

Oviatt, S. L. (1997). Multimodal interactive maps: Designing for human performance. Human-Computer Interaction (special issue on Multimodal Interfaces), 12, 93-129

Oviatt, S.L., Cohen, P.R., Wu, L.,Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. & Ferro, D. (2000). Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions, Human Computer Interaction. 15(4), 263-322

Oviatt, S. L., & Kuhn, K. (1998). Referential features and linguistic indirection in multimodal language. Proceedings of the International Conference on Spoken Language Processing, 6, 2339-2342. Syndey: ASSTA, Inc.

Rasmussen, J. (1980). What can be learned from human error reports? In M. K. Duncan, M. Gruneberg D. Wallis (Eds.), Changes in working life. New York:John Wiley & Sons.

Remington, R. (1999). Spoken language interface for a network management system. Proceedings of the International Society for Optical Engineering (SPIE) Multimedia Systems and Applications Conference (Boston, MA, September).

Remington, R. and Coven, H. (May 2002). Using a spoken language user interface for satellite command scripting. Proceedings of the 2002 Audio Input & Output Society Conference.

Reising, J.M. and Aretz, A.J. (1987). Color computer graphics in military cockpits. In H.J. Durrett(Ed.), Color and the Computer (pp 151-169). San Diego, CA: Academic Press, Inc.

Siebet, L.E. and Jacob, R.J.K. (2000). Evaluation of eye gaze interaction. CHI 200 Proceedings. 281-288.

Williamson, D.T. and Barry, T.P. (2000). The design and evaluation of a speech interface for generation of air tasking orders. Proceedings of the Human Factors and Ergonomics Society.

APPENDIX 1 - SATELLITE CONTROLLER TASK ANALYSIS

Table Of Contents

Analysis of Tasks Performed By Satellite Controllers

BACKGROUND

In order to determine how to best employ advanced human-system interface (HSI) technology to improve the performance of satellite controllers, it is necessary to understand the tasks performed by the controllers and to identify those areas where the controller's performance is hindered using the current system and technology. As part of the effort to meet these needs, a task analysis was performed.

The results of this task analysis are used to provide a baseline record of the actions performed by the controllers during an representative assortment of supports. This baseline is used to identify areas where HSI changes are likely to improve the performance of the controller, or to reduce the workload associated with performing particular tasks. HSI recommendations drawn from this task analysis are presented below.

The data used in the task analysis was obtained in October 2000 at the Air Force's Center for Research Support (CERES) facility at Shriever AFB, CO.

PARTICIPANTS

Two experienced satellite controllers from CERES participated in this task analysis. These controllers each had several years experience using the CERES satellite command and control system interface. Both had prior AF experience as satellite controllers.

BASELINE SYSTEM

The baseline satellite command and control system used in this task analysis is the COTS Based Real-Time Architecture (COBRA) system at CERES. COBRA is a proven system that is in day-to-day use at CERES to command and control satellites. COBRA consists of two workstations which communicate, but are not fully integrated. The controllers interact with the COBRA system using a mice and keyboards. Because the two workstations are not fully integrated, COBRA is configured with two keyboards and two mice. The controllers must switch between the left and right keyboard and mouse when they switch between the left and right workstations. Although COBRA has a "point and click" type of graphical interface, the telemetry screens consist of alphanumeric data, not graphical representations of the data.

The COBRA system is not used in any of the Space Operations Squadrons (SOPS), so far as we have been able to determine. The HSIs used in the SOPS appear to be text based, rather than graphics based, interfaces.

COBRA was selected to be the baseline for the task analysis as it appears to be the current state of the art among the HSIs implemented and used operationally by the Air Force, albeit to control Track and Command Only (TACO) satellites rather than operational satellites. There are several Commercial, Off-the-Shelf (COTS) HSIs that are more graphically oriented than is COBRA. CERES has assessed a number of these COTS HSIs, and has procured two systems that appear to offer performance improvements compared to COBRA. Neither of these COTS systems had

been fully integrated into the CERES facility at the time the task analysis data was collected, although integration was expected in the very near term.

PASS PLANS

A sample of the pass plans and tasks performed by controllers at CERES were selected for analysis. These tasks and pass plans were selected in consultation with CERES subject matter experts as covering most of the range of interactions between the controller and the system that are performed at CERES. Unfortunately, because of limitations in the ability to simulate satellite anomalies, anomaly resolution pass plans could not be included in this analysis.

Four categories of tasks were identified for inclusion in these analyses. These categories are:

(a) Pre-pass Tasks Performed Off-line
(b) Pre-pass Tasks Performed at the Controller's Workstation
(c) Pass Tasks Performed While in Contact With the Satellite
(d) Post-pass Tasks

<u>Pre-Pass Off-Line Task</u>

Schedule deconfliction a pre-pass task selected for analysis. This task is performed at least once per day. This task is performed after the daily schedule for use of Air Force Satellite Control Network (AFSCN) is published by 22 SOPS.

<u>Pre-Pass Tasks</u>

Prior to making contact with a satellite, controllers configure the command and control workstation and coordinate the resources necessary. Pre-pass procedures for contacts with IRONS 3160 and 9445 were analyzed. The pre-pass tasks are the same regardless of the satellite being supported, although some of the details (e.g., antenna pointing angles, power) will differ.

<u>Pass Plans</u>

A total of five supports, several requiring the controllers to perform multiple pass plans, were analyzed. These supports were:

- 3160 Set Battery A to FCAOA1 plus Momentum Estimation (MOMEST) plus Spin Control
- 3160 Impact Sensor
- 3160 Set Link 2 to 128K High plus Set Thermal Control System (TCS)to Circulate
- 9445 Eclipse Monitor plus Attitude Data Collection
- 9445 Health & Tracking plus Battery 1 Discharge Monitor plus No TLM procedure

All of the controllers task during a support are performed using the COBRA workstations.

Post-Pass Tasks

Two post-pass tasks were analyzed: The post-contact summary and updating the support schedule. These tasks are not performed at the command and control workstations used during the support.

DATA COLLECTION METHODS

Two methods of data collection were used in this effort. For tasks performed by the controller prior to configuring the controller's workstation and tasks performed following completion of the support (categories [a] and [d], above), an analyst recorded the actions of the controller in real-time. This record was discussed with the controller, and errors identified in this record were corrected.

For tasks performed at the workstation, either leading up to the point where the controller contacts the satellite or during contact with the satellite, videotape recordings were made for subsequent analysis. The majority of recordings were made using CERES' capability to simulate a support.

In order to capture the actions performed by the controller, the workstation's screens were videotaped. The workstation's screens were projected one at a time on a large, wall mounted screen. This projected image was actually videotaped, not the screen at the workstation. This approached eliminated the "scan lines" typical of recordings of CRT displays. The satellite controller selected the screen that was projected at each point in the simulated support. This selection was based on the task that was currently being performed, or what information was being examined. For example, when the controller was reading steps from the pass plan the left hand screen was projected. When the controller was examining values in the telemetry data, then the right hand screen was displayed. While switching screens required a few seconds, and added another task to those normally performed by the controller, the switching of screens in the videotape record improved the capability of the analyst to determine what the controller was doing, and what screen was being used to accomplish that task.

One of the on-line pre-pass tasks was video recorded during preparation for an actual support. In this case both the left and right screens are contained in each frame of the videotape. The videotape was not sufficient to allow identification of the control action being performed due to the small size of the screens given the resolution of the videotape. For the remainder of the tasks performed by the controllers at their workstations, the tasks were conducted using the satellite and systems simulation capability at CERES rather than being conducted during supports of actual satellites. This approach provided the opportunity to have the controllers elaborate on their actions verbally. It also allowed the controller to identify the screen being used at each point during the support. The commentary and the knowledge of the screen being attended to by the controller proved to be extremely useful when extracting information from the videotapes.

One undesirable side effect of this data collection and recording approach is that it makes it impossible to obtain objective measures of controller performance, particularly measures of task duration. This "cost" was considered to be acceptable in this instance because of limitations in the CERES simulation capability existing at that point in time. These limitations made it unlikely that the simulated system performance and telemetry would match what the controller

would see if actually performing a support with the prime systems. Therefore, it was expected that the controllers participating in the task analysis would unable to perform at the same tempo they would normally employ as they "worked around" limitations of the simulation. With this n mind, the an effort was made to provide rough order of magnitude estimates of the time that each task would take to perform under normal conditions. These estimates should be considered as being ordinal data at best, and caution should be taken when interpreting them.

DATA REDUCTION

While video recording the controllers, the analyst attempted to outline the tasks that were being performed. This was done by entering text into a spreadsheet. In cases where the analyst was unable to follow the actions of the controller, the analyst paused the simulation and queried the controller in greater depth. These initial entries were subsequently augmented with data extracted from the video tape records of the simulated supports.

Videotape recordings of the simulated supports were viewed by the analyst to extract the data needed to fill in the outlines created earlier. Each action performed by the satellite controller that was identified in the videotape was entered into a spreadsheet.

The product of this effort is a set of spreadsheets, one for each simulated pass plan.. The information in these spreadsheets identifies each action performed by the controller during the simulator support along with the modality or modalities required to perform those actions. Estimated times to complete each action are also provided. These times were generated by the analyst to provide a rough estimate of the relative duration required to complete each task, and are not based on objective performance measures collected during the simulated support. It is expected that the approximate durations would be useful in identifying tasks that are particularly time consuming.

RESULTS

Task analyses were prepared for the tasks described in Table 1. Table 1 also indicates the appendix that contains the task analysis detailing each of the tasks.

Table 1. Description of the contents of each of the task analyses.

SUPPORT PHASE	IRON	DESCRIPTION	APPENDIX
Pre-pass Performed Off-line	NA	Schedule Deconfliction	1
Pre-pass Tasks Performed at the Controller's Workstation	3160	Pre-pass	2
Pre-pass Tasks Performed at the Controller's Workstation	9445	Pre-pass	3
Pass Tasks Performed While in Contact With the Satellite	3160	Set Battery A to FCAOA1 plus Momentum Estimation (MOMEST) plus Spin Control	4
Pass Tasks Performed While in Contact With the Satellite	3160	Impact Sensor	5
Pass Tasks Performed While in Contact With the Satellite	3160	Set Link 2 to 128K High plus Set Thermal Control System (TCS) to Circulate	6
Pass Tasks Performed While in Contact With the Satellite	9445	Eclipse Monitor plus Attitude Data Collection	7
Pass Tasks Performed While in Contact With the Satellite	9445	Health & Tracking plus Battery 1 Discharge Monitor plus No TLM procedure	8
Post-pass Tasks	NA	Post Contact Scoring Summary	9
Post-pass Tasks	NA	Update Support Schedule	10

Number of Tasks Performed

A total of 921 individual tasks performed by the controllers were identified in the sample of pre-support, during support, and post-support tasks examined here. Table 2 shows the number of tasks performed in each of the supports.

Table 2. Number of Tasks Performed By the Controllers.

IRON	DESCRIPTION	TOTAL NUMBER OF TASKS PERFORMED BY THE CONTROLLER
NA	Schedule Deconfliction	29
3160	Pre-pass	137
9445	Pre-pass	137
3160	Set Battery A to FCAOA1 plus Momentum Estimation (MOMEST) plus Spin Control	236
3160	Impact Sensor	68
3160	Set Link 2 to 128K High plus Set Thermal Control System (TCS) to Circulate	64
9445	Eclipse Monitor plus Attitude Data Collection	140
9445	Health & Tracking plus Battery 1 Discharge Monitor plus No TLM procedure	83
NA	Post Contact Scoring Summary	8
NA	Update Support Schedule	19
	TOTAL	921

The differences in the total number of actions performed in each scenario reflect the fact that different numbers of pass plans were performed in the simulated scenarios ranged from one to three.

Modality Loading

Figure 1 shows the proportion of tasks in which the controller talked, listened, performed a manual task, performed a visual task, or performed a cognitive/decision making task.

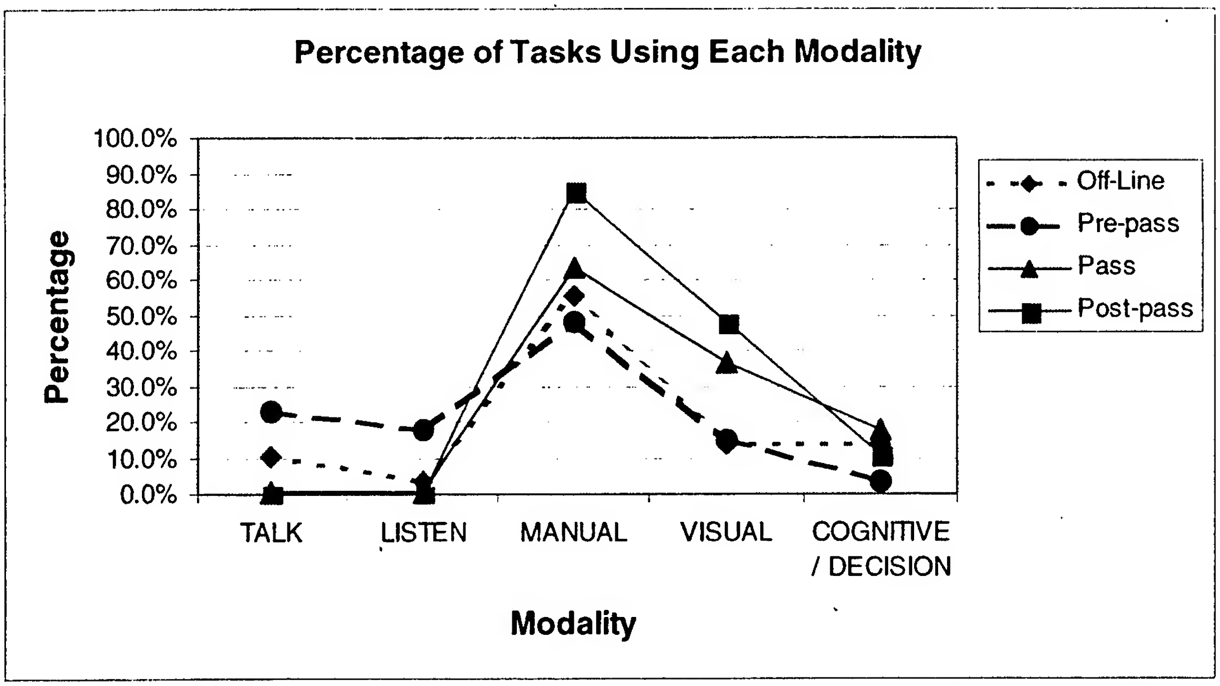


Figure 1. Proportion of tasks in which the controller talked, listened, performed a manual task, performed a visual task, or performed a cognitive/decision making task.

Inspection of this figure shows that for all categories of tasks considered in the tasks analyses, the bulk of the tasks require the controller to manually operate a control. Looking at the type of manual tasks performed, it is clear that the vast majority involve positioning the mouse and clicking the mouse button to select something. Other manual tasks, such as writing information down using a pen and paper or typing information in using the keyboard, are performed less frequently.

Closer examination of the data in the task listings shows that a very large number of mouse actions are needed to execute commands. In fact, to select a routine command in the pass plan and then to execute it requires four separate mouse clicks. If the command is restricted, then an additional mouse movement and an additional click are required. It is likely that the original reason for requiring multiple actions to cause a command to be executed was to allow the controllers (and possibly supervisors) to verify that the information entered by the controller was correct before the command was up-linked to the satellite. However, in COBRA the controller doesn't enter the command by typing it in; the command is already in the system and has been verified to be accurate. All the controller is doing is causing it to be sent. Therefore, in COBRA the need for a delay to allow verification no longer exists. The behavior of the controllers show that the controllers are not using the multiple mouse clicks to verify commands; all of the mouse clicks occur in rapid succession and no effort is made to perform any verification. (This isn't to say that the controllers are being negligent in any way. They simply don't have the need or opportunity to verify the accuracy of the command, which is essentially a set of approved macros.) This suggests that the need for multiple mouse clicks is no longer applicable and that the number of actions needed to cause a step in the pass plan to be executed could be reduced without increasing the probability of an erroneous command being sent.

Examination of the tasks involving manual action by the controller shows that a larger number of mouse clicks are made to select the appropriate telemetry screen on the right display. This indicates that the screens may not be arranged or organized in the best possible manner. When one looks at the screen it is clear that the alphanumeric presentation does not contribute to the controller's awareness of the relationships between the various values. Organizing the values so that their spatial arrangement is similar to the spatial arrangements of the components aboard the satellite may be one way to facilitate the controller's awareness.

Further inspection of Figure 1 shows that the next most commonly performed tasks make use of the controller's vision. These tasks generally involve comparing a value in the telemetred data with an expected value that is shown in the pass plan.

Of course, any use of the mouse has a visual component to it. Similarly, use of a keyboard involves the use of vision, at least for non-touch typists. Vision was not indicated as being central to either of those tasks when developing the task analysis spreadsheets. With this in mind, it is clear that the actual use of the visual modality is understated in this analysis. This suggests that allowing the controller to either gain more information in a single glance, co-locating information so that the controller does not have to search multiple locations to acquire , or transmitting information from the system to the controller through another modality (e.g., auditory) should be considered.

Performance Times.

As noted earlier, the time estimates contained in the task listings are estimates generated by the analyst. They are not based on objective measurement of the performance time in an operational setting. These estimates are only provided to provide a rough indication of the relative durations of the various tasks. It is likely that the time estimates are "worst case", and that the duration required by expert operators will be considerably shorter, particularly for manual tasks.

In general, the time required by the controllers to perform each of the tasks was only a few seconds. Even though there are a large number of tasks performed, the total times to perform the support are estimated to generally less than 10 minutes. However, some pass plans will require significantly longer to perform due to built-in waiting periods. The Momentum Estimation and Spin Control Maneuver, and the Battery Discharge pass plans, for example, contain steps which require the controller to simply wait while the effects of the commands take effect and the satellite reaches a new, stable state. These waiting periods range from about 2 minutes to as long as 20 minutes, and depending upon the results may be repeated.

HSI RECOMMENDATIONS

Keep Color Coding Accurate and Up-To-Date.

During each support controllers examine a variety of telemetry data to determine the satellite's state of health. Color coding of data is one method used to help the controllers identify data that is outside normal limits. This approach is only useful if the color coding is accurately applied. Unfortunately, a number of instances where the color coding is was not correct were observed

during the task analysis simulations and on other observations conducted at CERES. This problem may be unique to a facility like CERES where the satellites have outlived their design life, often by significant lengths of time. An example of a typical problem on such a satellite is that consumable resources (e.g., thruster propellant) have been depleted. Such a low level in an operational satellite would likely be a problem that should be brought to the controller's attention by coding the value in red, indicating a critical level. However, the use of red to indicate low levels of consumables in the case of extremely old satellites tends to diminish the ability of color coding to capture the controller's attention; they learn to ignore the color coding. (This may be less of a problem in settings where the satellites are still operational.)

Automate Comparisons

Comparisons between expected values contained in the pass plan and the values contained in the telemetered data should be automated. During the pass, the majority of tasks classified as being cognitive/decision involve comparing a data value in the telemetry with a value, or range of values, contained in the pass plan. These simple comparisons should be automated, and the results of the comparisons displayed in a manner that allows the controller to quickly scan and approve or disapprove continuing. If continuing is disapproved then the controller would make a decision regarding the proper manner to proceed. This could range from selecting an anomaly procedure to be performed, or contacting others to resolve the problem.

Provide Tools to Aid in Identifying Out-of-Tolerance Values.

The controllers currently assess the state of the satellite during each support. This requires examination of several pages of telemetry data, and a comparison of the observed with expected values. In order to maximize the detection of out of tolerance values and to reduce the controller's workload, the task of comparing each data item with its nominal, warning, and caution values should be allocated to the computer rather than the human. This tool could be implemented as either a continuously running application, or as an application launched by the controller. The benefit of a continuously running application is that when the controller examines the page (or window) containing the list of out of tolerance values, they would be assured that they are seeing the most current data. The potential drawback to this approach is that transient out-of-tolerance conditions caused by the controller issuing a command would be added to the list. If, on the other hand, the application was launched by the controller then transients would not appear on the list.

Provide Tools to Assess the Effects of Selected Actions.

Some commands require the controller to wait a period of time before assessing the effect of that command. The waiting period is somewhat arbitrary, and is usually somewhat longer than would be optimally efficient. (A MOMEST procedure is an example where there is a 20 minute wait.) The use of controller time and of system resources would be more efficiently used if the system evaluates the convergence of the parameter to a new set point, and alerts the controller when that criterion is reached.

Provide Tools to Allow the Controller to "Drill Down" Into the Data.

The controllers should have the capability to "drill down" and examine related data for each summary value displayed. This capability should be implemented so that it can be used from a page showing all out-of-tolerance values, as well as from any point where the controller has access to the data values.

Activating this tool would cause a display containing the value of the particular variable to be displayed, along with the values of other variables that effect the variable. For example, if a battery temperature is out-of-tolerance (either shown in color in a display or listed as being out-of-tolerance) the controller should be able to activate this tool. When activated, the drill-down tool would bring up a page showing the exact voltage of the battery, its voltage range during that session, and the states or values of other satellite components that might cause the battery's voltage to be high or low. The other satellite components that might be on such a page depends on the specifics of the satellite, but one might expect the state of the thermal control system in the region of the battery to be included, as well as measures of electrical system state such as current draw, charging voltage, and the like.

This tool should allow the controller to access the history of the variables over multiple contacts. Continuing with the example of a battery temperature, the controller should be able to call up a display showing the trend of the battery over the last *N* contacts or over the last *N* days or weeks (where *N* is an integer defined by the controller). These displays should provide the controller with tools to help define and interpret the trend. The tools should not necessarily be restricted to computing and displaying the linear component of any trend; higher order functions may also be needed, depending on the system.

Reduce the Number of Manual Tasks Required to Perform a Step in the Pass Plan.

The number of actions that must be performed to execute a command or to access the data of interest should be kept to the minimum. The controllers must click the left mouse button a total of four times (five if the command is restricted) to execute a step in the pass plan. This large number of actions may be a legacy from earlier systems, where each mouse click allowed the action to be cross-checked. However, in COBRA this is no longer true; the controllers simply make multiple mouse clicks seemingly without pausing between them. An additional step is warranted when it provides the controller an opportunity to verify that the action is in fact the action intended, particularly when the action could have a serious adverse impact on the satellite if performed incorrectly or at the wrong time. Restricted commands are one example where verification of intent is appropriate.

Eliminate the Need to Use Multiple Keyboards and Multiple Mice.

A single keyboard and a single mouse (or any other cursor controller) should allow the controller to interact with any of the displays at the workstation. The current layout of the COBRA system requires two keyboards and two mice. The left hand keyboard and mouse allow the controller to interact with the display on the left side of the workstation, and the right ones allow the controller to interact with the display on the right side of the workstation. The controllers are continually switching back and forth between these tools to accomplish their task.

Provide Alternative Methods of Interacting with the Workstations.

Other methods of effecting the displays on the workstation should be employed. As can be seen in Figure 1, the vocal and auditory modalities are virtually unused during a contact. This suggests that voice recognition and voice synthesis are likely candidates for exploitation as interface methods.

Voice Recognition and Synthesis.

Voice recognition would be useful as a means for controlling the workstation. Almost any command that can effect the display of data could be implemented as a voice activated command. Examples include, but are not limited to:

- Sending commands (e.g., "send SARM", "execute next step [in the pass plan]", "reset impact sensor")
- Displaying data (e.g., "show battery temperatures", "show residual momentum")
- Changing the way data is displayed (e.g., "show on thermometer", "show digital value", show trend over the last week")

Voice synthesis could be used to alert the controller to specific events of interest, or to provide feedback on the satellite's status. Examples of an alerting functions range from announcing that a variable has changed state (e.g., "battery 1 is overheated", "valve 1 is now closed") to cueing the controller that it is time to perform a task (e.g., "twenty minutes has elapsed, examine the residual momentum value", "it is now time to perform the spin correction procedure").

The use of voice recognition and synthesis will reduce the dependency on manual actions. It also has the potential to reduce the controller's cognitive workload. This reduction would come from eliminating the need for the controller to maintain a cognitive map of the location of telemetry values which is used to find the values of interest. For example, if the controller needs to examine a value, they currently "click" on the button that calls up the desired telemetry page. If they forget where the variable is located, they end up searching though the pages. Using a voice recognition system, they would simply state that they want to see a particular variable and the system would locate and display that information.

Touchscreens.

Touchscreens provide an alternative to other cursor control devices, such as mice, track balls, joysticks, and force buttons for interacting with the workstations. Touchscreens are a mature technology that in some instances offer performance improvements over mice and other cursor control devices. Conditions where touchscreens are generally superior include:

- Movement of the cursor over a large screen distance would be required
- Extremely fine cursor positioning is not required

Touchscreens are not a panacea. They often require a layer of material between the observer and the display surface. This can degrade the optical quality of the display. Furthermore, the use of touchscreens can result in smudging of the display, and a consequent loss of image clarity. In some cases this can reduce display legibility below acceptable levels. Another shortcoming of

touchscreens is they usually do not offer an input mode that corresponds directly to clicking the center or right mouse buttons (assuming a right handed mouse configuration).

3-D Glove.

A 3-D glove would be an adjunct to the existing manual interfaces. A 3-D glove would allow the controller to perform tasks such as moving virtual switches from one position to another. It would also allow the controller to "grab" the image of the satellite or its systems and rotate the image so as to get a different perspective. With 2-D and perspective views of the satellite, this function is likely to be adequately supported by a touch screen or by more conventional cursor controls.

If a true stereoscopic image was presented, then a 3-D glove would allow the controller to select and manipulate virtual items in that image. The need for a stereoscopic display has not been identified for the controller tasks examined here. However, such displays may be a useful control device supporting future needs such as non-autonomous robotic servicing of satellites.

Other Input Devices.

The range of other cursor control devices that could be considered for the satellite control workstation of the future is unlimited. However, we have not identified a technology that both offers a performance improvement over the devices mentioned above and, in our judgement, is sufficiently mature to warrant including in the prototype at this point in time. However, we would like to mention eye tracking briefly here. Eye tracking techniques may allow the system to determine what the controller is looking at each moment in time. This could conceivably be used in lieu of other cursor control devices. However, the performance claims of existing COTS eye tracking systems coupled with the head movement limitations and other constraints, leads us to believe that they are not yet suitable as an input device.

Although these systems are not currently adequate as input devices to be used by controllers, we see this technology as being a potentially valuable objective measure of the focus of the controller's visual attention in a research and evaluation setting. Knowledge of where the controller is looking, and how long the controller looks at that location, provides an indication of how effectively information is extracted from a particular display. Alternative displays can be compared in terms of fixation frequency and duration.

<u>Improve the Organization of Telemetry Data Displays.</u>

Data that is needed or accessed by the controller at the same time should be located on a single page, and should be adjacent. With the current system, values that are often inspected at the same step are not in close proximity, and may even be on different pages.

Adaptive organization of data displays is one approach that should be considered. The grouping of variables would be tailored to support the controller accomplish the immediate task. This requires the control workstation to know or infer the intent of the controller, and the information needed to satisfy that intent. For example, if the controller was verifying battery voltages in the spacecraft, then a page would be displayed in which data from all of the batteries is displayed. If the controller was examining the effects of temperatures in a portion of the satellite, then the

battery voltage(s) from the part of the satellite being examined would be displayed along with other temperature sensitive measures.

With the COBRA system, some variables are listed on multiple pages. While this may require the controller to memorize the multiple locations in order to find a value when it is needed, it does have the potential to limit the need for the controller to call up multiple pages in order to view all of the data items of interest. Unfortunately, the current organization of the telemetry pages does not group the data based on the needs of the controller at that point in the support. Therefore, the controller often is required to switch pages to see all of the data of interest.

Normalize Data Values.

In order to allow controllers to support multiple types of satellites more easily, normalized values should be presented rather than raw values. Currently, data pages the raw values of variables. The values that are within tolerance often differ between satellites. Since the raw values are relatively unimportant in terms of assessing satellite health, the values should be presented in normalized form. For example, the voltages and temperatures of batteries should be 100% when nominal. This would allow the controller to quickly determine if any values are out-of-tolerance without requiring reference to a specific value shown in a pass plan, and without requiring the controller to memorize and recall the values that are appropriate for that satellite. (Using raw values can be a bigger problem when the ideal values are easily confused, for example 33.6 degrees and 36.3 degrees as opposed to 33.6 degrees and 12.0 degrees. Use of nominal values avoids this problem entirely.)

Fuse Satellite State of Health Data into a Summary Display.

Data from the satellite should be fused into a single summary display area. The summary should indicate the overall state of health of the satellite at that point in time, and indicate where trends in the state of health should be examined by the controller. The goal of this display is to allow the controller to quickly determine if all systems are nominal, and if not, which systems require attention. Currently, the controller identifies out of tolerance systems by examining the raw telemetry pages manually; the system does not provide support for detection of problems other than color coding of telemetry values.

Auditory alerting of changes in the satellite's state of health during a contract, or from the last contact, should accompany the visual display. The type of auditory alert needs to be considered further. Two options are a tone and voice alerts. A tone is simple to generate, and would be used to call the controller's attention to the display. At that point the controller would examine the display and make a determination of what systems(s) is in an anomalous state. A voice warning could be used to provide the controller additional information about the nature of the problem, or even advice on how to proceed.

Positioning of Data Scanned Continuously by the Controller.

Data that is scanned consistently during a support, the once per second data indicator for example, should be grouped together and positioned in a dedicated, readily visible screen location. This will allow the controllers to develop and use a scan pattern. A structured scan

pattern will increase the probability that a controller will be able to accurately and quickly identify changes in the system's status that effects the ability to perform the support.

Provide Indication of System Problems.

The task analysis shows that the controller is not extensively aided in evaluating the state of the systems used to perform the satellite support. To better support the controller we recommend that the controller be alerted when built in testing detects a problems or conditions that could effect the accuracy of the telemetry. This will require development of system tests which run continuously in the background. (These are sometimes referred to as Continuous Built-In Tests [CBIT].) For example, the system should monitor the data coming from the primary and alternate communications paths. When the data is not identical, then there is a possibility that the data being observed and evaluated by the controller is not accurate; the workstation may be displaying the "good" data, or it may be displaying the "bad" data which could be mistaken as indicating a problem with the satellite. Alerting the controller that the data from the two sources isn't the same would allow the controller to more accurately make a determination if there is a real problem with the satellite, or if the observations are anomalous because of problems unrelated to the satellite. Making this determination accurately and quickly could allow a critical support to continue, rather than having the controller mis-use time diagnosing the source of the problem.

Provide an Estimate of the Time Required to Perform Each Procedure.

The estimated time required to perform each procedure should be made available to the controller, and cross checked against the remaining support time. In the event of an anomaly, the controller makes a determination of what procedures need to be run, if any, to rectify the problem. In some cases, there may not be enough time left in the scheduled support to accomplish the anomaly procedure, or to accomplish the procedure without eliminating other support objectives. In order to make good decisions on how to proceed, the controller needs to be aware of the expected duration of the anomaly procedure, and which of the remaining tasks can be completed. It may well be that the controller decides to request an extension to the support. In this case, providing a realistic estimate of the length of extension required would allow more efficient scheduling of AFSCN assets. Alternatively, the controller may elect to either postpone performance of the anomaly procedure, or may elect to drop lower priority tasks from the support. In either case, knowing the expected duration will allow the controller to make better decisions regarding time allocation during the support.

ACKNOWLEDGEMENTS
This work could not have been completed without the assistance of CERES. Special thanks are given to Mr. Charles Jackson and Mr. Carlo Maldonado for sharing generously of their time and expertise, and for arranging use of equipment needed to simulate and video record the supports reported here.

APPENDIX 1 - DECONFLICTION

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEED BACK	NOTES
	Boot System												
1.1.1		Type in user ID			x					keyboard	0:00:15		
1.1.2		Type in Password			x					keyboard	0:00:15		
1.1.3		Press the ENTER key in response to ready prompt			x					keyboard	0:00:02		
	Select Action												
1.2.1		Click on the "Deconfliction" menu option			x					mouse (keyboard alternate)	0:00:05	screen change	
1.2.2		Click on the "Run Macro" menu option			x					mouse (keyboard alternate)	0:00:05	screen change	
1.2.3		Click on the "Deconflict" menu option			x					mouse (keyboard alternate)	0:00:05	screen change	
	Select contact to be deconflicted												
1.3.1		Type in the time scheduled for the event to be deconflicted			x					keyboard	0:00:15		
	System												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEED BACK	NOTES
	action												
1.4.1		Wait for printer									0:00:30	hardcopy ejected from printer	
	Document												
1.5.1		Walk over to printer and get printout			x						0:00:10		
1.5.2		Put printout into the PAP binder			x						0:00:15		
	Examine Printout for Satellite A												
1.6.1		Compare requested time with time allocated				x	x				0:01:00		time from 1 to 5 minutes, depending on the number of conflicts and changes
1.6.2		Pencil in new times if they are in the acceptable window as defined in the PAP									0:00:15		time per change. This is repeated for each change
	Examine Printout for Satellite B												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEED BACK	NOTES
1.7.1		Compare requested time with time allocated				x	x				0:01:00		time from 1 to 5 minutes, depending on the number of conflicts and changes
1.7.2		Pencil in new times if they are in the acceptable window as defined in the PAP									0:00:15		time per change. This is repeated for each change
	Examine Printout for Satellite C												
1.8.1		Compare requested time with time allocated				x	x				0:01:00		time from 1 to 5 minutes, depending on the number of conflicts and changes
1.8.2		Pencil in new times if they are in the acceptable window as defined in the PAP									0:00:15		time per change. This is repeated for each change
	Examine Printout for Satellite D												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEED BACK	NOTES
1.9.1		Compare requested time with time allocated				x	x				0:01:00		time from 1 to 5 minutes, depending on the number of conflicts and changes
1.9.2		Pencil in new times if they are in the acceptable window as defined in the PAP									0:00:15		time per change. This is repeated for each change
1.10	Call 22 SOPS												
1.10.1		Phone 22 SOPS, ask for person to do deconfliction	x	x							0:00:30		
1.10.2		Identify change of interest	x								0:00:30		repeated for each change
1.10.3		Accept or reject change proposed by 22 SOPS	x								0:00:10		repeated for each change
1.10.4		End phone call											
	Create hardcopy confirming schedule changes												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEED BACK	NOTES
1.11.1		Click on "Create MSC"			x					mouse (keyboard backup)	0:00:05		time per item
1.11.2		Enter MSC			x					mouse (keyboard backup)	0:00:30		
1.11.3		Close Window			x					mouse (keyboard backup)	0:00:20		
1.11.4		Select "Send MSG"			x					mouse (keyboard backup)	0:00:05		
1.11.5		Select receipt to be printed			x					mouse (keyboard backup)	0:00:15		
1.11.6		Print receipt			x						0:00:15		
1.11.7		Pick up receipt from printer			x						0:00:10	hardcopy ejected from printer	

APPENDIX 2 - PRE-PASS – IRON 3160

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1	Logon to both workstations												Solaris operating system on the COBRA system at CERES
1.1.1		Type in the USER ID on the left workstation			x					kbd	0:00:15		
1.1.2		Press the ENTER key			x					kbd	0:00:05	Cusor moves to next field	
1.1.3		Type in the PASSWORD on the left workstation			x					kbd	0:00:15		
1.1.4		Press the ENTER key			x					kbd	0:00:05	Window requesting USERID & Password is removed from the screen and the booting process continues	
1.1.5		Type in the USER ID on the right workstation			x					kbd	0:00:15		
1.1.6		Press the ENTER key			x					kbd	0:00:05	Cusor moves to next field	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.7		Type in the PASSWORD on the right workstation			x					kbd	0:00:15		
1.1.8		Press the ENTER key			x					kbd	0:00:05	Window requesting USERID & Password is removed from the screen and the booting process continues	
1.2	Check system status												
1.2.1		Look at ESD to identify known system outages that could impact the ability to communicate with the satellite				x	x			Log book	0:00:30		Controller must know how outage will effect plans for the current contact and, if there are outages, the "work arounds"
1.3	Start User Interface												
1.3.1		Click on UI (icon on the lower left of the screen) on			x					mouse, left display	0:00:05		this brings up the Main Control Panel on the left

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		the left workstation.											workstation
1.3.2		Click on the REAL TIME SESSION button			x					mouse, left display	0:00:05		
1.3.3		Scroll down until VEH_9445_OP is found			x					mouse, left display	0:00:10		
1.3.4		Click on VEH_9445_OPS on the pull down menu			x	x				mouse, left display	0:00:05	the only indication of a system hang up is that the messages don't update.	
1.3.5		Click OK			x					mouse, left display	0:00:05	Two windows will appear when this is done; Alarm, Warnings & Errors (AWE) and Real Time Session Executive Panel (Main Control Panel, MCP)	
1.4	Set up session												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.4.1		Click File Iln the Real-Time Session window			x					mouse, left display	0:00:05	pop up window appears	
1.4.2		Click on OPEN CONFIG			x					mouse, left display	0:00:05		
1.4.3		Click OK			x					mouse, left display	0:00:05	pop up window is removed from the screen	
1.5	Verify Session Set Up												
1.5.1		Look at the Alarm, Warnings & Errors (AWE) window and see that you get the DOWNLOAD COMPLETE FOR ALL DIRECTORIES message and no error messages				x			AWE window	Left display	0:00:10	no time out or failure messages	
1.6	Start SYS500												
1.6.1		Click on FILE in the Realtime Session window			x					mouse, left display	0:00:05	a pop up window appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.6.2		Click on OPEN CONFIGURATION			x					mouse, left display	0:00:05	A list appears	
1.6.3		Click on 2-START-APPS-SYS00.CFG so it is highlighted			x					mouse, left display	0:00:05	Selected item is highlighted	
1.6.4		Click OK			x					mouse, left display	0:00:05	Eleven applications will load. A double asterisk (**) indicates loading is in process	
1.6.5		Verify process completes				x				Left display	0:01:30	look for SYSTEM 500 SETUP COMPLETE message	
1.7	Minimize G2 wirdow												
1.7.1		Click on minimize icon in the window			x					mouse, left display	0:00:05		the G2 window is launched at this point. You don't need it yet so minimize it
1.7.2	Verify load completed prcperly												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.7.3		Look for VEHICLE CONFIGURATION OF G2 IS COMPLETE message				x				Left display	0:00:15		G2 = GynSym
1.8	Define contact information												
1.8.1		Click on FILE in the REALTIME SESSION window			x					mouse, left display	0:00:05		
1.8.2		Click on SEND CONTACT ID			x					mouse, left display	0:00:05		
1.8.3		Type in IRON			x					kbd	0:00:15		
1.8.4		Type in STATION (all capital letters)			x					kbd, left display	0:00:15		
1.8.5		Type in CONTACT START TIME			x					kbd, left display	0:00:15		
1.8.6		Type in REV			x					kbd, eft display	0:00:15		
1.8.7		Click on OK			x					mouse, left display	0:00:05		
1.8.8		Verify the data you just typed in is correct				x	x		REAL TIME SESSION PANEL	Left display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.8.9		Click OK			x					mouse, left display	0:00:05		
1.9	Select string												
1.9.1		Click on the CONNECT TO AIM button (upper right of the screen)			x					mouse, left screen	0:00:05	pop up window appears	
1.9.2		Click on the name of the string that is to be used			x					mouse, left screen	0:00:05	String intered into field	
1.9.3		Click OK			x					mouse, left screen	0:00:05		
1.9.4		Click FILE in the executive Session Window			x					mouse, left display	0:00:05		
1.9.5		Click OPEN CONFIGURATION			x					mouse, left display	0:00:05		
1.9.6		Click on 3-START-APPS-POST-SYS500.CFG so it is highlighted			x					mouse, left display	0:00:05	Selected item is highlighted	
1.9.7		On the REALTIME SESSION WINDOW click			x					mouse, left display	0:00:05		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		APPLICATIONS											
1.9.8		Click MONITOR			x					mouse, left display	0:00:05		
1.9.9		Observe/verify applications launch				x	x						
1.9.10		Click OK if all applications lauch successfully			x					mouse, left display	0:00:05		
1.9.11		Click APPLICATIONS in the REALTIME SESSION			x					mouse, left display	0:00:05		
1.9.12		Click LAUNCH			x					mouse, left display	0:00:05		
1.9.13		Click on the applications desired so that they are highlighted			x					mouse, left display	0:00:05		
1.9.14		Click OK. This will open the DATA GROUP PANEL			x					mouse, left display	0:00:05		
1.9.15		Click on TRACKDATA on the DATA GROUP			x					mouse, left display	0:00:05	Selection becomes highlighted	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		PANEL so it is highlighted											
1.9.16		Type in TRKDATA			x					mouse, left display	0:00:15		
1.9.17		Click SEND			x					mouse, left display	0:00:05		
1.9.18		Click to maximize the G2 window			x					mouse, left display	0:00:05		
1.9.19		Click on MODES			x					mouse, left display	0:00:05		
1.9.20		Click on ADV-OPERATOR			x					mouse, left display	0:00:05		
1.9.21		Verify SYBASE green				x			IMT window		0:00:15		IMT = Intelligent Missionn Toolkit
1.9.22		Verify MESSAGE green				x			IMT window		0:00:15		
1.9.23		Verify AP-ARITH green				x			IMT window		0:00:15		
1.9.24		Click on CONTACT in the IMT window			x					mouse, left display	0:00:05		
1.9.25		Click on EXECUTE COMMAND PROCEDURE			x					mouse, left display	0:00:05		
1.9.26		Scroll down to desired Pass Plan			x					mouse, left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.9.27		Click OK			x					mouse, left display	0:00:05		
1.9.28		Click on the down arrow to make the step current OR click on the step you want to make the current step			x					mouse, left display	0:00:05		
1.9.29		Click on GO AUTOMATIC SETUP			x					mouse, left display	0:00:05		
1.9.30		Monitor to verify that all steps show COMPLETE								Left display	0:00:30		
1.9.31		Verify that the time is the same and correct on both workstation screens and on the clock mounted in the console				x				Left and right displays, clock	0:00:15	If there is a problem, the list will freeze and you will get a "Validation Failed" message on the screen	
1.9.32		Click ARTS on the ACM panel			x					mouse, left display	0:00:05		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.9.33		Click RGF so it is highlighed			x					mouse, left display	0:00:05		RGF = Remote Ground Facility
1.9.34		Click OK			x					mouse, left display	0:00:05		
1.9.35		Verify that the ARTS selected is correct				x					0:00:15		
1.10	Load Two Line Element (TLE)												
1.10.1		Click TLE on the ACM panel			x					mouse, left display	0:00:05		TLE generates the antenna pointing angles that will be given to the ARTS for acquisition
1.10.2		Click OPEN			x					mouse, left display	0:00:05		
1.10.3		Click ont IRON.TLE			x					mouse, left display	0:00:05		
1.10.4		Click OK			x					mouse, left display	0:00:05		
1.10.5		Verify year and date on TLE are within about 30 days of the current date									0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.10.6		Click SHOW APA PROFILE			x					mouse, left display	0:00:05		This shows the Antenna Pointing Angles. These are needed for the briefing. You can leave the window up or write the angles down
1.11	Make phone contacts to set up communications networks												
1.11.1		Call Falcon Control (FC)	x	x						phone	0:00:10		
1.11.2		Wait for reply from FC		x						phone		Falcon control answers the phone	Time is varaible
1.11.3		Brief FC (Satellite IRON, ARTS to be used, communication resources required, data rates expected)	x						A standard briefing outline is available. The controller may use this and pre-enter the information in grease pencil	phone	0:02:00		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.11.4		Listen to readback from Falcon Control		x			x			phone	0:01:00		
1.11.5		Confirm readback accuracy	x							phone	0:01:00		Any things that FC read back incorrectly must be corrected by the controller
1.11.6		Call ARTS	x	x						phone	0:00:10		
1.11.7		Wait for ARTS to acknowledge								Phone		ARTS operator answers the phone	Time is variable
1.11.8		Brief ARTS (Controller name, IRON, IRON configuration [default is common], start and end support times.)	x							Phone	0:01:30		
1.11.9		Request system status, readback of system time (in UTC down to seconds) and name of ARTS operator.	X							Phone	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.11.10		Verify time given by ARTS operator matches time at the console		x		x				digital clock on console	0:00:20		Controller makes sure that the time (UTC) matches the time as shown on the digital clock within two or three seconds
1.11.11		Confirm to ARTS operator that you understood name and system status, and that the time is correct.	x							Phone	0:00:10		Ususally just say "copy" to confirm
1.11.12		Tell ARTS operator that the Unclassified briefing is coming	x							Phone	0:00:10	ARTS operator responds confirming ready for briefing	
1.11.13		Provide unclassified briefing (antenna pointing angles [azimuth &, elevation in degrees], expected range in	x			x			procedures binder for that satellite	Phone, written briefing outline in the procedures book for that satellite	0:02:00		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		nautical miles, when to go active [King's directive], when to enable uplink mod, SGLS channels ,uplink power, lost comm procedure to use)											
1.11.14		Request readback	x							Phone	0:00:10		
1.11.15		Listen to readback from the ARTS operator		x		x					0:01:00	Controller listens to readback and compares the values with those read from the procedures book	
1.11.16		Confirm accuracy of readback	x			x				Phone	0:00:10		If the readback is not correct then the controller must correct the ARTS operator.
1.11.17		Tell ARTS operator to stand by for prepass	x							Phone	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.12	Get feedback from FC regarding resource avaiability												
1.12.1		Listen for call from FC that resources are configured as requested		x						Phone			The timing of this call depends on how long it takes FC to set up their part of the system
1.12.2		Confirm that you heard call from FC reporting resources are appropriately configured	x							Phone	0:00:10		Simply say "Roger FC" or "Copy FC" to acknowledge that controller heard them confirm that they have their part of the system set up
1.13	Connect Software												
1.13.1		Set up the software so that you can later SARM the ARTS			x					Mouse, left display	0:00:30		This appears to be about 4 or 5 mouse clicks on the left screen
1.14	SARM												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.14.1		Ask ARTS if they are ready for a SARM	x							phone	0:00:10		SARM is an acronym for "Set Asynchronous Response Mode"
1.14.4		Listen for reply from ARTS.		x						phone		ARTS operator reports that they are ready to SARM	This is usually quick as the ARTS operator is already on the phone
1.14.5		Click on button to SARM			x					mouse, left display	0:00:05		
1.14.6		Verify that communications are established as expected				x				Left display	0:00:15	once per second data should begin to increment	
1.14.7		Confirm to ARTS that SARM has been completed and that you have communications	x							Phone	0:00:10		
1.14.8		Call FC	x							Phone	0:00:10		
1.14.9		Wait for FC to acknowledge		x						Phone		FC answers the phone	Time can be variable. It depends on what other tasks FC is performing

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.14.10		When FC acknowledges tell them that you have connection	x							Phone	0:00:10		In this case, all that came up was narrow band. Controller asked FC to look into why there wasn't a wide band connection
1.15	No wide bard connection problem resolution (In this case, only the narrow bard communication link between ARTS and the controller's workstation)												
1.15.1		Call FC	x							Phone	0:00:10		
1.15.2		Listen for FC to come up on the phone		x						Phone		FC answers the phone	Time is variable
1.15.3		Ask FC to look and see if they are getting	x							Phone	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		data on the wide band CCS port											
1.15.4		FC responds affirmative or negative		x						phone	0:00:10	FC reports whether or not they are getting data from DSCS	FC may have the controller standby while making this determination
1.15.5		Call ARTS	x							Phone	0:00:10		
1.15.6		Listen for ARTS to come up on the phone		x						Phone		ARTS operator answers the phone	Time is variable
1.15.7		Ask ARTS to verify that they are in "CSOC" ("see - sock") node	x							Phone	0:00:10		
1.15.8		Listen for FC or ARTS to report back after they check their systems		x						Phone	0:00:10	ARTS operator responds affirmatively or negatively	
1.15.9		Tell FC that you are going to "SARM again"	x							Phone	0:00:10		This action depends on what problem, if any, is reported by FC

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.15.10		FC acknowledges that they are ready to receive the SARM command		x						Phone	0:00:10		
1.15.11		Issue SARM command			x					mouse, left display	0:00:05		
1.15.12		Report results of action performed by controller to FC	x							Phone	0:00:10		In this case, re-SARMing did not solve the problem
1.15.13		Listen for FC's response to report		x						phone	0:00:10	FC reports whether or not they are gettng dara after the re-SARM	
1.16	Continue doing support using only narrowband												
1.16.1		Call ARTS to get them up on the voice system	x							Phone	0:00:10		
1.16.2		Listen for ARTS to be on the voice system		x						phone		ARTS operator answers the phone	Time can be variable

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.16.3		Instruct ARTS to configure for contact and to acquire the satellite	x							Phone	0:00:10		
1.16.4		Listen for ARTS to confirm that they have acquired the vehicle		x						Phone	0:00:10	ARTS operator either confirms communications with the satellite, or reports that they are not communicating	there can be some time variability depending on how fast the satellite is tracked
1.16.5		Observe that you are getting expected data				x	x			Left display	0:00:15	Satellite range will be shown, and the once per second data will increment. Range and APAs should be close to the expected values briefed by the controller earlier	
1.17	Cotiue troublesho oting the lac< of wiceband												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
	connection												
1.17.1		Listen for FC to call with report on what they have found or have done to fix the problem		x						Phone		FC calls with status and summary of efforts done to resolve problem	Time depends on what FC does to rectify the problem, and the success that they have.
1.17.2		Confirm you heard FC's report	x							Phone	0:00:10		
1.17.3		Provide FC with report on the status of the ongoing contact ("We have a good lock on the 128K")	x							Phone	0:00:20		
1.17.4		Listen for FC to report next action they took to resolve problem		x						Phone			Time depends on what FC does to rectify the problem, and the success that they have. In this case FC changed a patch cord

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.17.5		Negotiate with FC regarding next action to be taken to resolve the problem	x	x						Phone	0:01:00		How much work to be done to resolve the problem varies with the problem. In this case the controller decided to just use the narrow band link for the support
1.17.6		Call ARTS to let them know that the support will be done using the backup link	x							phone	0:00:10		
1.17.7		Listen for ARTS operator to acknowledge		x						phone		ARTS operator acknowledges use of the backup link instead of primary link	Time can be variable
1.18	Configure system to cortinue the support												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.18.1		Click to send antenna pointing angles to ARTS			x					mouse (multiple clicks)	0:00:30		about 4 mouse clicks on the left screen
1.18.2		Monitor display to verify that the angles are being sent out from the controller's workstation and are being used by the ARTS t point the antenna				x				left display	0:00:45	If antenna angles are sent successfully, it indicates that the communications link between the ARTS and the controller's workstation is working.	
1.19	Cortinue troubleshooting lack of widaband												
1.19.1		Call FC to get them on the voice system	x							Phone	0:00:10		
1.19.2		Listen for FC to be on the voice system		x						Phone		FC answers the phone	Time is variable, depending on what other tasks FC is doing

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.19.3		Ask FC to see if they are receiving wideband data from the ARTS (This required FC to hook up an oscilliscope to their equipment so that they can visually determine if they are receiving data from the ARTS)	x							Phone	0:00:10		This would indicate whether the problem is between ARTS and FC, or between FC and the Controller's workstation. The controller uses his or her knowledge of the AFSCN to guide the actions taken to troubleshoot the problem.
1.19.4		Listen for FC to tell you that they are ready to monitor communications		x						Phone		FC reponds by announcing that they are ready to monitor the communications electronically.	It may take a few minutes for FC to connect the equipment needed to monitor the communicaitons (ie., an ocilliscope).

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.19.5		Click to send SARM command when FC is ready			x					mouse, left display	0:00:05		
1.19.6		Listen for FC feedback on result of sending SARM		x						Phone		FC reports wheter they saw a change in communicaitons as a result of the SARM being resent	In this case, it appears that the problem was between FC and the ARTS.
1.20	Continue support using available assets												
1.21.1		Examine telemetry for data outside of allowable ranges				x				right display	0:01:00		A high battery temperature was found on examination. The controller was so familiar with the nominal values that the out of tolerance value was recognized. However, because the color coding is

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													not kept current, the controller can't simply scan for values that are red or yellow.

APPENDIX 3 - PRE-PASS – IRON 9445

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1	Logon to both workstations												Solaris operating system on the COBRA system at CERES
1.1.1		On left workstation enter USERID			x					kbd, left workstation	0:00:15		
1.1.2		Press the ENTER key			x					kbd, left workstation	0:00:05	Cursor moves to the next field	
1.1.3		On left workstation enter PASSWORD			x					kbd, left workstation	0:00:15		
1.1.4		Press the ENTER key			x					kbd, left workstation	0:00:05	System begins booting	
1.1.5		On right workstation enter USERID			x					kbd, right workstation	0:00:15		
1.1.6		Press the ENTER key			x					kbd, right workstation	0:00:05	Cursor moves to the next field	
1.1.7		On right workstation enter PASSWORD			x					kbd, right workstation	0:00:15		
1.1.8		Press the ENTER key			x					kbd,right workstation	0:00:05	system begins booting	
1.2	Check system status												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.1		Look at ESD to identify known system outages that could impact the ability to communicate with the satellite				x	x				0:00:30		Controller must know how outage will effect plans for the current contact and, if there are outages, the "work arounds"
1.3	Start User Interface												
1.3.1		Click on UI (icon on the lower left of the screen) on the left workstation			x					mouse, left workstation	0:00:05		this brings up the Main Control Panel on the left workstation
1.3.2		Click on the REAL TIME SESSION button			x					mouse, left workstation	0:00:05	A pop up menu appears	
1.3.3		Scroll down and select VEH_9445_OPS									0:00:10	the only indication of a system hang up is that the messages don't update.	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.4		Click on VEH_9445_OPS			x	x				mouse, left workstation	0:00:05		
1.3.5		Click OK			x					mouse, left workstation	0:00:05	Two windows will appear when this is done; Alarm, Warnings & Errors (AWE) and Real Time Session Executive Panel (Main Control Panel, MCP)	
1.4	Set up session												
1.4.1		Click FILE in the Real-Time Session window			x					mouse, left display	0:00:05	A pop up window appears	
1.4.2		Select OPEN CONFIG			x					mouse	0:00:05		
1.4.3		Click OK			x					mouse	0:00:05	The pop up window is removed from the display	
1.5	Verify Session Set Up												

			MODALITY										
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.5.1		Look at the Alarm, Warnings & Errors (AWE) window and see that you get the DOWNLOAD COMPLETE FOR ALL DIRECTORIES message and no error messages				x			AWE window		0:00:20	no time out or failure messages	
1.6	Start SYS500												
1.6.1		Click FILE in Realtime Session window			x					mouse, left display	0:00:05		
1.6.2		Click on OPEN CONFIGURATION			x					mouse, left display	0:00:05		
1.6.3		Highlight 2-START-APPS-SYS00.CFG			x					mouse, left display	0:00:05		
1.6.4		Click OK			x					mouse, left display	0:00:05	Eleven applications will load. A double asterisk (**) indicates loading is	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
												in process	
1.6.5		Verify process completes				x				Left display	0:00:30	look for SYSTEM 500 SETUP COMPLETE message	
1.6.6	Minimize G2 window												
1.6.1		Click on minimize icon in the window			x					mouse, left display	0:00:05		the G2 window is launched at this point. You don't need it yet so minimize it
1.7	Verify load completed properly												
1.7.1		Look for VEHICLE CONFIGURATION OF G2 IS COMPLETE message				x				Left display	0:00:15		G2 = GynSym
1.8	Define contact information												
1.8.1		Click FILE in the REALTIME			x					mouse, left display	0:00:05		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		SESSION window											
1.8.2		Click on SEND CONTACT ID			x					mouse, left display	0:00:05		
1.8.3		Type in IRON			x					kbd, left display	0:00:15		
1.8.4		Type in STATION (all capital letters)			x					kbd, left display	0:00:15		
1.8.5		Type in CONTACT START TIME			x					kbd, left display	0:00:15		
1.8.6		Type in REV			x					kbd, left display	0:00:15		
1.8.7		Click on OK			x					mouse, lef display	0:00:05		
1.8.8		Verify the data you just typed in is correct				x	x		REAL TIME SESSION PANEL	Left display	0:00:15		
1.8.9		Click OK			x					mouse, left display	0:00:05		
1.9	Select string												
1.9.1		Click on the CONNECT TO AIM button (upper right of the screen)			x					mouse, left display	0:00:05		
1.9.2		Select string that is to be used			x					mouse, left display	0:00:05		

TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MODALITY MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.9.3		Click OK			x					mouse, left display	0:00:05		
1.9.4		Click FILE in the executive Session Window			x					mouse, left display	0:00:05		
1.9.5		Click OPEN CONFIGURATION			x					mouse, left display	0:00:05		
1.9.6		Highlight 3-START-APPS-POST-SYS500.CFG			x					mouse, left display	0:00:05		
1.9.7		Click APPLICATIONS on the REALTIME SESSION WINDOW			x					mouse, left display	0:00:05		
1.9.8		Click MONITOR			x					mouse, left display	0:00:05		
1.9.9		Observe/verify applications launch				x	x			Left display			
1.9.10		Click OK if all applications lauch successfully,			x					mouse, left display	0:00:05		
1.9.11	-	Click APPLICATIONS in REALTIME SESSION			x					mouse, left display	0:00:05		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.9.12		Click LAUNCH			x					mouse, left display	0:00:05		
1.9.13		Highlight applications desired			x					mouse, left display	0:00:05		
1.9.14		Click OK.			x					mouse, left display	0:00:05	DAT GROUP PANEL window opens	
1.9.15		On the DATA GROUP PANEL highlight TRACKDATA			x					mouse, left display	0:00:05		
1.9.16		Type in TRKDATA			x					kbd, left display	0:00:15		
1.9.17		Click SEND			x					mouse, left display	0:00:05		
1.9.18		Click to maximize the G2 window			x					mouse, left display	0:00:05		
1.9.19		Click on MODES								mouse, left display	0:00:05		
1.9.20		Click on ADV-OPERATOR			x					mouse, left display	0:00:05		
1.9.21		Verify SYBASE green				x			IMT window	Left display	0:00:05		IMT = Intelligent Mission Toolkit
1.9.22		Verify MESSAGE green				x			IMT window	Left display	0:00:05		
1.9.23		Verify AP-ARITH green				x			IMT window	Left display	0:00:05		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.9.24		Click on CONTACT in the IMT window			x					mouse, left display	0:00:05		
1.9.25		Click on EXECUTE COMMAND PROCEDURE			x					mouse, left display	0:00:05		
1.9.26		Scroll down to desired Pass Plan			x					mouse, left display	0:00:05		
1.9.27		Click OK			x					mouse, left display	0:00:05		
1.9.28		Click on the down arrow to make the step current OR click on the step you want to make the current step			x					mouse, left display	0:00:05		
1.9.29		Click on GO AUTOMATIC SETUP			x					mouse, left display	0:00:05		
1.9.30		Monitor to verify that all steps show COMPLETE				x				Lleft display	0:00:30		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.9.31		Verify that the time is the same and correct on both workstation screens and on the clock mounted in the console				x				Left and right displays, clock	0:00:15	If there is a problem, the list will freeze and you will get a "Validation Failed" message on the screen	
1.9.32		Click ARTS on the ACM panel			x					mouse, left display	0:00:05		
1.9.33		Highlight RGF			x					mouse, left display	0:00:05	RGF name entered into field	RGF = Remote Ground Facility
1.9.34		Click OK			x					mouse, left display	0:00:05		
1.9.35		Verify that the ARTS selected is correct				x				Left display	0:00:05		
1.10	Load Two Line Element (TLE)												

TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MODALITY MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.10.1		Click TLE on the ACM panel			x					mouse, left display	0:00:05		TLE generates the antenna pointing angles that will be given to the ARTS for acquisition
1.10.2		Click OPEN			x					mouse, left display	0:00:05		
1.10.3		Select IRON.TLE			x					mouse, left display	0:00:05		
1.10.4		Click OK			x					mouse, left display	0:00:05		
1.10.5		Verify year and date on TLE are within about 30 days of the current date				x				Left display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.10.6		Click SHOW APA PROFILE			x					mouse, left display	0:00:05		This shows the Antenna Pointing Angles. These are needed for the briefing. You can leave the window up or write the angles down
1.11	Make phone contacts to set up communications networks												
1.11.1		Call Falcon Control (FC)	x	x						phone	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.11.2		Wait for reply from FC		x						phone		FC acknowledges call	Time is too variable to estimate reliably. Range is from a few seconds to several minutes. Controller may have to try to make contact several times depending on what else FC is doing.
1.11.3		Brief FC (Satellite IRON, ATRS to be used, communication resources required, data rates expected)	x						A standard briefing outline is available. The controller may use this and pre-enter the information in grease pencil	phone	0:01:30		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.11.4		Listen to readback from ARTS		x			x			phone	0:00:20	ARTS operator reads back the key parts of the briefing.	The controller listens and identifies any errors in the read back
1.11.5		Confirm readback accuracy	x							phone	0:00:10		Any things that FC read back incorrectly must be corrected by the controller
1.11.6		Call ARTS	x	x						phone	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.11.7		Wait for ARTS to acknowledge		x						phone		ARTS operator comes up on the phone	Time is too variable to estimate reliably. Range is from a few seconds to several minutes. Controller may have to try to make contact several times depending on what else the ARTS operator is doing.
1.11.8		Brief ARTS (Controller name, IRON, IRON configuration [default is common], start and end support times.)	x							phone	0:01:30		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.11.9		Request system status, readback of system time (in UTC down to seconds) and name of ARTS operator.	x								0:00:10		
1.11.10		Listen to the system status report from ARTS operator, noting anything that could effect the support		x						phone	0:00:30		
1.11.11		Verify time given by ARTS operator matches time at the console		x		x				digital clock on console	0:00:15	ARTS operator reads time and the controller compares it against the time shown on the workstation's clock. If it is off by more than a couple of seconds there is a problem.	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.11.12		Confirm to ARTS operator that you understood name and system status, and that the time is correct.	x							phone	0:00:10		Ususally just say "copy" to confirm
1.11.13		Tell ARTS operator that the Unclassified briefing is coming	x							phone	0:00:10	ARTS operator indicates readiness to receive briefing	
1.11.14		Provide unclassified briefing (antenna pointing angles [azimuth &, elevation in degrees], expected range in nautical miles, when to go active [King's directive], when to enable uplink mod, SGLS channels ,uplink power,	x			x				Phone. Written briefing outline is in the procedures book for that satellite	0:01:30		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		lost comm procedure to use)											
1.11.15		Request readback	x							Phone	0:00:10		
1.11.16		Listen to readback. Compare values in readback with expected values on briefing page		x		x	x			Phone	0:01:00	ARTS operator reads back key points of briefing	If the readback is not correct then the controller must correct the ARTS operator.
1.11.17		Confirm accuracy of readback	x							Phone	0:00:10		
1.11.18		Tell ARTS operator to stand by for prepass	x							Phone	0:00:10	ARTS operator confirms stanfing by	
1.12	Get feedback from FC regarding resource avaiability												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.12.1		Listen for call from FC that resources are configured as requested		x						Phone		FC announces that the configuration is as requested by the controller	The timing of this call depends on how long it takes FC to set up their part of the system
1.12.2		Confirm that you heard call from FC reporting resources are appropriately configured	x							Phone	0:00:10		Simply say "Roger FC" or "Copy FC" to acknowleddge that you heard them tell you that they have their part of the system set up
1.13	Connect Software												
1.13.1		Set up the software so that you can later SARM the ARTS			x					mouse, left display	0:00:45		This appears to be about 4 or 5 mouse clicks on the left screen
1.14	SARM												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.14.1		Ask ARTS if they are ready for a SARM	x							phone	0:00:10		
1.14.2		Listen for reply from ARTS.		x						phone		ARTS operator announces that they are ready for the controller to send the SARM command	Time is too variable to estimate. It depends on what the ARTS operator is doing
1.14.3		Click on button to SARM			x					mouse, left display	0:00:05		
1.14.4		Verify that communications are established as expected				x				Left display	0:00:10	once per second data should begin to increment	
1.14.5		Confirm to ARTS that SARM has been completed and that you have communications	x							Phone	0:00:10		
1.14.6		Call FC	x							Phone	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.14.7		Wait for FC to acknowledge		x						Phone		FC verbal acknowledgment	Time too variable. Depends on what other tasks FC is doing
1.14.8		When FC acknowledges tell them that you have connection	x							Phone	0:00:10		In this case, all that came up waas narrow band. Controller asked FC to look into why there wasn't a wide band connection
1.15	No wide band connection problem resolution (In this case, only the narrow band communication link between ARTS and the controller's workstatio												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
	n)												
1.15.1		Call FC	x							Phone	0:00:10		
1.15.2		Listen for FC to come up on the phone		x						Phone		FC annunces that they are on the phone line	Time is too variable to estimate. Depends on FC
1.15.3		Ask FC to look and see if they are getting data on the wide band CCS port	x							Phone	0:00:10		
1.15.4		Call ARTS	x							Phone	0:00:10		
1.15.5		Listen for ARTS to come up on the phone		x						Phone		ARTS operator announces that they are on the phone line	Time is too variable to estimate. Depends on ARTS operator
1.15.6		Ask ARTS to verify that they are in "CSOC"	x							Phone	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		("see - sock") node											
1.15.7		Listen for FC or ARTS to report back after they check their systems		x						Phone		Verbal status report	Time is too variable to estimate. Depends on how fast FC and ARTS operator can check their systems
1.15.8		FC reports ... (unintelligable)		x						Phone	0:00:10	FC verbal communications	
1.15.9		Tell FC that you are going to "SARM again"	x							Phone	0:00:10		This action depends on what problem, if any, is reported by FC
1.15.10		Listen for FC to indicate that they are ready for the SARM command		x						phone		FC announces that they are ready for the controller to send the SARM command	Variable time. Normally just a few seconds

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.15.11		Click to issue SARM command			x					mouse, left screen	0:00:05		
1.15.12		Report results of action performed by controller to FC	x							Phone	0:00:10		In this case, re-SARMing did not solve the problem
1.15.13		Listen for FC's response to report		x						Phone	0:00:10	FC gives verbal report	FC will be on the phone when SARM command is sent
1.15.14													1
1.16	Continue doing support using only narrowband												
1.16.1		Call ARTS to get them up on the voice system	x							Phone	0:00:10		
1.16.2		Listen for ARTS to be on the voice system		x						Phone		ARTS operator comes on the phone	Variable time
1.16.3		Instruct ARTS to configure for contact and to acquire	x							Phone	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		the satellite											
1.16.4		Listen for ARTS to confirm that they have acquired the vehicle		x						Phone		ARTS operator verbally confirms contact	Variable time. Depends on how much the ARTS operator has to do to configure and how long it takes to make contact with the satellite
1.16.5		Observe that you are getting expected data				x				Left display	0:00:15	Satellite range will be shown, and the once per second data will increment	
1.17	Contiue troubleshooting the lack of wideband connection												
1.17.1		Listen for FC to call with report on what they have		x						Phone		FC verbally reports on efforts made and	Variable time

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		found or have done to fix the problem										results	
1.17.2		Confirm you heard FC's report	x							Phone	0:00:10		
1.17.3		Provide FC with report on the status of the ongoing contact ("We have a good lock on the 128K")	x							Phone	0:00:10		
1.17.4		Listen for FC to report next action they took to resolve problem		x						Phone		FC verbally reports on efforts made and results	Variable time. (FC changed a patch cord in this case)

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.17.5		Negotiate with FC regarding next action to be taken to resolve the problem	x	x						Phone			Variable time. How much work to be done to resolve the problem varies with the problem. In this case the controller decided to just use the narrow band link for the support
1.17.6		Call ARTS to let them know that the support will be done using the backup link	x							Phone	0:00:10		
1.18	Configure system to continue the support												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.18.1		Send antenna pointing angles to ARTS			x					mouse	0:00:15		about 4 mouse clicks on the left screen
1.18.2		Monitor display to verify that the angles are being sent out from the controller's workstation and are being used by the ARTS t point the antenna				x				left display	0:00:20	Ifantena angles are sent successfully, it indicates that the communications link between the ARTS and the controller's workstation is working.	
1.19	Continue troubleshooting lack of wideband												
1.19.1		Call FC to get them on the voice system	x							Phone	0:00:10		
1.19.2		Listen for FC to be on the voice system		x						Phone		FC announces that they are on the phone	Variable time

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.19.3		Ask FC to see if they are receiving wideband data from the ARTS (This required FC to hook up an oscilliscope to their equipment so that they can visually determine if they are receiving data from the ARTS)	x							Phone	0:00:10		This would indicate whether the problem is between ARTS and FC, or between FC and the Controller's workstation. The controller uses his or her knowledge of the AFSCN to guide the actions taken to troubleshoot the problem.

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.19.4		Listen for FC to tell you that they are ready to monitor communications		x						Phone			Variable time. It depends on what FC needs to do to comply with the controller's request
1.19.5		Send SARM command when FC is ready			x					mouse	0:00:10		
1.19.6		Listen for FC feedback on result of sending SARM		x						Phone		FC reports results verbally	Variable time. In this case, it appears that the problem was between FC and the ARTS.
1.20	Continue support using available assets												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.21		Examine telemetry for data outside of allowable ranges				x				right display			A high battery temperature was found on examination. The controller was so familiar with the nominal values that the out of tolerance value was recognized. However, because the color coding is not kept current, the controller can't simply scan for values that are red or yellow.

APPENDIX 4 - SET BATTERY A TO FCAOA1 PLUS MOMENTUM ESTIMATION (MOMEST) PLUS SPIN CONTROL – IRON 3160

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1	Finished SOH and ready to begin Batt-A-FCAOA1 pass plan												When editing this file, down towards the end there were lots of manually selecting and executing a command. This requires 3 clicks. The first click makes the command active. The second prepares it to be sent (downloads) and the third executes the command
1.1.1		Click on CONTACT on IMT menu (on the top of the left display) at end of the SOH bring up the BATT-A-FCAOA-1 pass plan			x					mouse, left display	0:00:05	A drop down menu appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		by .											
1.1.2		Click on SELECT COMMAND PROCEDURE in the drop down menu			x					mouse, left display	0:00:05	a pop up window appears. This is the window that allows the controller to select a pass plan from among those that have already been created	
1.1.3		Click on down arrow to bring up depository of pass plans (a list of the procedures that the MC can run)			x					mouse, left display	0:00:05		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.4		Click on the BATTERIES-AUTO item in the submenu. This is where the controller expects the desired pass plan to be located.			x	x				mouse, left display	0:00:05	A pop up window containing the pass plans related to BATTERIES-AUTO appears to the left of the window	Location of pass plans is by visual search or controller's memory
1.1.5		Visually locate the desired pass plan (BATTERT A to FCAOA1 in this case)				x				Left display	0:00:05		
1.1.6		Click on desired pass plan			x					mouse, left display	0:00:05	The fields in the pass selection window are filled with the information from the selected item and the pop up windows used to select the plan are removed from the display	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.7		Click on OK button in the Pass Plan Depository window			x					mouse, left display	0:00:05		
1.1.8		Read instructions. This pass plan (Batt-A-FCAOA1) is run normally before entering eclipse				x				Left display	0:00:10		
1.1.9		Click on SNAP button in the lower left hand corner of the right screen. This will bring of screen full of acronyms and the associated TLM			x					mouse, right display	0:00:05	The SNAP screen appears in the right hand display	
1.1.10		Visually inspect the information in the SNAP frame to identify any data that is obviously out of				x	x	Knowledge of nominal values and limits OR accurate visual coding of TLM data	TLM data on the right display	Right display	0:00:15		The controller has to know what the expected values of the data are to make this inspection, or the

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		tolerance											color coding needs to be up to date.
1.1.11		Go back to pass plan and verify data points listed on the pass plan by reading the values off the SNAP window (6 values are verified going back and forth between the left and right screens)				x	x	Nominal values and limits for each variable of interest. These are contained in the pass plan.	Pass Plan contains nominal values and limits, TLM page contains current values	Alternate between the left display(read the Pass Plan) and the right display (read the value of the variable)	0:01:00		The controller typically reads one line from the pass plan, and then locates and reads the value from the SNAP screen. Six (6) values are verified at this point
1.1.12		Go back to Pass Plan and read that you are to send a command by clicking on it to make it active				x				Left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.13		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.1.14		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.1.15		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.1.16		Verify that data is being received at the once per second rate				x	x	knowledge that satellite is transmitting/receiving data	Changing indicator	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.1.17		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.18		Verify that the change occurred by reading the value on the SNAP screen				x	x	Controller to verify that the value or state changed as expected	Pass Plan contains nominal values and limits, TLM page contains current values	Right display	0:00:15	Controller reads the value from the SNAP page to verify that the expected change has been made and is reflected in the telemetry	This is odd in the sense that what the controller is looking at ar ethe first letters of a series of items. When correct, the letters are FCAOA (Full, Connect, Auto, Open, Auto)
1.1.19		Go back to pass plan - there is an instruction to notify SE if a value exceeds a criterison. (If EPB A Temp > X degrees then notify SE)				x	x	Current value and criterion value	Criterion on Pass Plan. The current value is on the TLM screen	Left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.20		Go to SNAP screen and check value. If <> criterion call SE else end pass plan				x	x	Current value and criterion value	Criterion on Pass Plan. The current value is on the TLM screen	Right display	0:00:15		Controller compares value called out in the pass plan with the TLM value in the SNAP screen
1.1.21		Click on the endpost in the task plan			x					mouse, left display	0:00:05	Returns the left display to the SOH plan at the point where another pass plan can be selected	
	end of BATTERY A to FCAOA pass plan												
1.2	Begin MOMEST pass plan - MOMEST is a procedure found in operations checklist. MOMEST is embedded in the SPIN CONTROL Pass plan												This is normally a process in which the controller does a momentum estimation, then a spin control maneuver, and then another momentum estimation

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													to see that the spin control maneuver had the desired effect.
1.2.1		Click on the Real Time Session window. This is outside the pass plan window			x					mouse, left display	0:00:05		If the Real Time Session window isn't visible, then the controller may have to resize or minimize the pass plan window
1.2.2		Click on APPLICATIONS in the real time session application panel (which is on the left display) .			x					mouse, left display	0:00:05	A pull down window will appear.	

			MODALITY										
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.3		Click on LAUNCH in the pull down menu			x					mouse, left display	0:00:05	A window with a list of procedures will appear.	
1.2.4		Scroll down until the desired procedure (s) are found in the window.			x					mouse, left display	0:00:10		The desired functions may be outside the viewable area. If so, the controller needs to scroll the window until they can be seen. (They were outside the area in this example).
1.2.5		Click on LINK-1 FETUNIX			x					mouse, left display	0:00:05		
1.2.6		Click on the Display Host button			x					mouse, left display	0:00:05	A pull down menu containing the names of the available strings appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.7		Click on the name of the correct Display Host System in the scroll down menu			x	x				mouse, left display	0:00:05	Field in the window is filled in with the name of the string to be used	
1.2.8		Click on the Execution Host button			x					mouse, left display	0:00:05	A pull down menu containing the names of the available strings appears	
1.2.9		Click on the name of the correct Execution Host system in the scroll down menu			x	x				mouse, left display	0:00:05	Field in the window is filled in with the name of the string to be used	
1.2.10		Click on LAUNCH			x					mouse, left display	0:00:05		
1.2.11		Verify that the procedure is launching by inspecting the Applications Status Panel window. This				x	x	Controller to verify that normal execution is underway.	Observation of changes in the window showing the exection of the steps.	Left display	0:00:15	In another window on this display, text showing the steps the system is performing to launch the applicatio	The controller doesn't have any tools to aid in monitoring the execution of these steps.

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		window shows the execution of the steps										n are displayed.	
1.2.12		Scroll down until LINK-1 RECEIVE PACKET is visible.			x	x				mouse, left display	0:00:10		
1.2.13		Click on LINK-1 RECEIVE PACKET			x					mouse, left display	0:00:05		
1.2.14		Click on Display Host button			x					mouse, left display	0:00:05	A pull down menu containing the names of the available strings appears	
1.2.15		Click on the name of the correct Display Host System in the scroll down menu			x	x				mouse, left display	0:00:05	Field in the window is filled in with the name of the string to be used	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.16		Click on the Execution Host button			x					mouse, left display	0:00:05	A pull down menu containing the names of the available strings appears	
1.2.17		Click on the name of the correct Execution Host from pull down menu			x	x				mouse, left display	0:00:05	Field in the window is filled in with the name of the string to be used	
1.2.18		Click on LAUNCH			x					mouse, left display	0:00:05		
1.2.19		Verify launched in the Applications Status panel				x	x	Indication that the process is launched and proceeding normally	Progress shown in the Application Status Panel	Left display	0:00:15	In another window on this display, text showing the steps the system is performing to launch the application are displayed.	The controller doesn't have any tools to aid in monitoring the execution of these steps.
1.2.20		Scroll down until LINK-1 MOMEST is visible in the window			x	x				mouse, left display	0:00:10		
1.2.21		Click on LINK-1 MOMEST			x					mouse, left display	0:00:05		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.22		Click on the Display Host button			x					mouse, left display	0:00:05	A pull down menu containing the names of the available strings appears	
1.2.23		Click on the name of the correct Display Host from pull down menu			x	x				mouse, left display	0:00:05	Field in the window is filled in with the name of the string to be used	
1.2.24		Click on the Execution Host button			x					mouse, left display	0:00:05	A pull down menu containing the names of the available strings appears	
1.2.25		Click on the name of the correct Execution Host from pull down menu			x	x				mouse, left display	0:00:05	Field in the window is filled in with the name of the string to be used	
1.2.26		Click on OK			x					mouse, left display	0:00:05	The window is closed.	This is a click on OK, not launch. OK both launches the procedure and closes the window

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.27		Verify MOMENST has launched in the Applications Status panel				x	x	Indication that the process is launched and proceeding normally	Progress shown in the Application Status Panel	Left display	0:00:15	In another window on this display, text showing the steps the system is performing to launch the application are displayed.	The controller doesn't have any tools to aid in monitoring the execution of these steps.
1.2.28		Click on the Pass Plan icon to make it the visible window (that is, bring it to the front). The icon is in the lower left of the screen			x					mouse, left display	0:00:05	The Pass Plan window appears in front of all the other windows	
1.2.29		On the right display click on the LINK1 DISPLAY button			x					mouse, right display	0:00:05		This isn't described in the pass plan. The controller needs to know which TLM display page to bring up.

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.30		Click on MOMEST in the Link 1 display page			x					mouse, right display	0:00:05		
1.2.31		Watch the line graph to see if the value is moving towards zero and becomes a constant value				x	x	Controller examines spin trend to verify that the change is in the expected direction and is convergin g on the nominal rate	Graph on the right display showing spin rate error. (This should converge towards zero.)	Right display	0:20:00	TLM converges towards the "0.0" line on the graph.	Value is usually positive to start. The controller watches the graph too see that the trend is convergin g towards the desired value.
1.2.32		Print MOMEST results by clicking on the wall paper in the right screen			x					mouse, right display	0:00:30		
1.2.33		Click on Print Window			x					mouse, right display	0:00:05	a pop up window with print options appears	
1.2.34		Click on Reverse			x					mouse, right display	0:00:05	cursor changes from arrow to a cross hair when ready to select the window to be printed	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.35		Click on window you want printed with the cross hair cursor			x					mouse, right display	0:00:05	When the screen to be printed is selected a tone is generated and the cursor changes back to an arrow	
0	end MOMEST												
1.3	Begin SPIN CONTROL												
1.3.1		On the pass plan, you will be at the point in the State of Health (SOH) pass plan that tells you to entier select and execute another pass plan or to exit. In this example, a MOMEST and spin control will be performed								Left display			

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.2		Read instruction in the Pass Plan. This instruction tells the controller to select the ACS DELTA V MANEUVER.				x				Left display	0:00:15		
1.3.3		Click on LINK 2 DISPLAYS on the right display			x					mouse, right display	0:00:05	Brings up a page of buttons on the rightt display	
1.3.4		Click on ACS DELTA V MANEUVER button on the right display			x					mouse, right display	0:00:05	a page of data mneumics appears	(ACS DELTA V MANEUVER is the name of a page that will be displayed)
1.3.5		Go to pass plan and read instructions				x				Left display	0:00:15		
1.3.6		Verify SUNREF=PRA on the ACS DELTA V MANEUVER page (on the right screen)				x	x	Nominal and current values of this variable	Nominal is in the Pass Plan, current is on the right display	Right display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.7		Verify SPLN = A0 (an "A" followed by a zero) every 10 seconds) on the ACS DELTA V MANEUVER page				x	x	Nominal and current values of this variable	Nominal is in the Pass Plan, current is on the right display	Right display	0:01:00		time is approximate. The problem is if it isn't updating
1.3.8		Go back to pass plan and click on continue arrow			x					mouse, left display	0:00:05	A new page of the pass plan comes up	
1.3.9		Read instructions on pass plan				x				Left display	0:00:15		
1.3.10		Verify the five values listed on the pass plan. The TLM is on the ACS DELTA V MANEUVER page.				x	x	Nominal and current values of these variables	Nominal values are in the Pass Plan, current values are on the right display	Right display	0:00:15		scan back and forth. If a wrong value tell SE who would be sitting next to you
1.3.11		Go back to pass plan and read. It tells you to select SNAP frame				x				Left display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.12		Click on the SNAP frame button on the right display			x					mouse, right display	0:00:05	The SNAP frame display replaces the previous frame	
1.3.13		Read instructions on Pass Plan. It tells you to verify a value on the SNAP page				x				Left display	0:00:15		
1.3.14		Verify that the value on the SNAP screen is within tolerance				x	x	The allowable limits and current value of this variable	The limits are in the Pass Plan, current is on the right display	Right display	0:00:15		if value out of tolerance there is a procedure to run per the pas splan
1.3.15		Go back to pass plan and read instruction				x				Left display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.16		Wait until the time listed in the pass plan to continue				x	x	Current time and the time when the command is to be executed	Current time is on a digital clock at each workstation, the time to execute the command is precomputed and listed in the pass plan	Clock		Monitor the digital clock on the workstation	The SE and OA have tailored the pass plan before the support. They have entered a time for the start of any spin control commands to be sent
1.3.17		Click on down arrow at the designated time			x					mouse, left display	0:00:05		The time is on digital clocks located to the left of both screens at each workstation. All times are Zulu
1.3.18		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.19		Click on the "MAKE CURRENT" item in			x					mouse, left display	0:00:05	An arrow appears to the left of the command	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		the pop up window											
1.3.20		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.21		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:15	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.22		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.23		Read the instruction on the pass plan. It has the mission planner (or SE) enter a data value into the pass plan				x				Left display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.24		Verify CMDREG value is the same as designated by the mission planner				x	x	The allowable limits and current value of this variable	The limits are in the Pass Plan, current is on the right display	Right display	0:00:15		
1.3.25		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.26		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.27		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.28		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.29		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.30		Verify the values of DELE and DELO on SNAP page Orverify the values on the LINK 2 DISPLAY page. (These variables are listed redundantly on at least these two pages)				x	x	The allowable limits and current values of these variables	The correct values are in the Pass Plan, the current values are on the right display	Right display	0:00:15		the values are displayed on at least these two pages. The format for the displays and the locations of the data are different on the two pages
1.3.31		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.32		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.33		Click on the arrow to the left of the command			x					mouse, left display	0:00:05	Arrow turns pink	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		to make it active											
1.3.34		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.35		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.36		Verify value of ASJEBB (= enabled) on ACS DELTA V MANEUVER screen				x	x	Current and desired values of this variable	Desired value is in the pass plan and the current value is in the TLM data	Right display	0:00:15		
1.3.37		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.38		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.39		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.40		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.41		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.42		Verify CMDREG value on ACS DELTA V MANEUVER frame				x	x	Current and desired values of this variable	Desired value is in the pass plan and the current value is in the TLM data	Right display	0:00:15		
1.3.43		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.44		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.45		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.46		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:15	data is observed to be changing at about once per second, indicating that the satellite is communic	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
												ating	
1.3.47		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.48		Verify INHIBIT value on ACS DELTA V MANEUVER page				x	x	Current and desired values of this variable	Desired value is in the pass plan and the current value is in the TLM data	Right display	0:00:15		
1.3.49		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.50		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.51		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.52		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.53		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.54		Verify CMDREG value on ACS DELTA V MANEUVER page				x	x	Current and desired values of this variable	Desired value is in the pass plan and the current value is in the TLM data	Right display	0:00:15		
1.3.55		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.56		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.57		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.58		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per seco+J11 2nd data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.59		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.60		Verify SUBMOD value on the ACS DELTA V MANEUVER page				x	x	Current and desired values of this variable	Desired value is in the pass plan and the current value is in the TLM data	Right display	0:00:15		
1.3.61		go back to pass plan page and read instructions				x					0:00:15		
1.3.62		Record value of DSPIN. You have to go back to ACS DELTA V MANEUVER page and read the value. The value is written on scratch paper			x	x				paper & pencil, right display	0:00:30	write it down	This value will be used in some later calculations. The MC writes it down on a piece of scatch paper
1.3.63		Go back to the pass plan and read the instructions. This tells you to record the value				x				Left display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		of PPLN1P											
1.3.64		Click on the SNAP screen button on the right screen			x					mouse, right display	0:00:05	SNAP screen replaces the DELTA V MANEUVER page on the right display	Controller needs to know which screen has the desired data. The screen isn't identified in the Pass Plan.
1.3.65		Find PPLN1P and record the value on the scratch paper			x	x				Pencil & paper	0:00:30		
1.3.66		Go back to the pass plan and read instructions				x				Left display	0:00:15		
1.3.67		Click on the LINK 2 DISPLAY button in the Right display			x					mouse, right display	0:00:05	A screen with a series of buttons appears on the right screen	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.68		Click on ACS DELTA V MANEUVER button (on the LINK 2 DISPLAY page) to bring up that page			x					mouse, right display	0:00:05	The ACS DELTA V MANEUVER screen appars	
1.3.69		Click the down arrow in the pass plan to continue			x					mouse, left display	0:00:05	The next page of the Pass Plan appears on the screen	
1.3.70		Read the instruction and follow it. You are doing a simple calculation to check on the SE's computations			x	x	x	spin rate, plenum pressure	TLM page	Pencil & paper or calculator	0:00:15		Computation of the number of firings. This is done by the controller and the SE. If they both get the same number, then they assume that the computations were done correctly.
1.3.71		Click on CONFIGURE COMMANDING on the left			x					mouse, left display	0:00:05	pop up window appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		display											
1.3.72		Click on REPEAT AS NEW REQUEST			x					mouse, left display	0:00:05	Button infront of the option selected is filled in. The buttons are hollow circles if not selected	
1.3.73		Click on SET MODE PARAMETERS button to bring up next window			x					mouse, left display	0:00:05	a pop up window ("REPEAT MODE PARAMETERS") is displayed	
1.3.74		Click on the box with number of times command is to be sent In the REPEAT MODE PARAMETERS window			x					mouse, left display	0:00:05	The frame of the box is highlighted, indicating that the controller can type a number into the box	The number of times to repeat the command was computed by the controller and the SE
1.3.75		Type in the number of times you want to send the command			x					kbd	0:00:10	the number appears in the box as it is typed in	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.76		Press the ENTER key on the keyboard when the number has been entered.			x					kbd	0:00:02	Once the ENTER key is pressed, the highlighting is removed from around the box.	
1.3.77		Click on the DONE button			x					mouse, left display	0:00:05	The pop up window is removed from the screen	
1..3.78		Verify on CONTACT CONFIGURATIONS button that the number of transmissions is the same as the value just typed in				x	x	Number of transmissions intended and number performed	The number of transmissions performed is on the Pass Plan display. The number intended is from the computations performed by the controller and SE	Left display	0:00:15		Controller reads the number and makes sure it matches the number of repetions just typed in.

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.79		Click close window box			x					mouse, left display	0:00:05	The pop up window is removed from the display	This is a small box with an "X" in it. The box is located in the upper left hand portion of the pop up window.
1.3.80		Go back to pass plan, if the rotation rate is fast follow one decision path, if slow another. The MC and SE compare the actual values with desired values to make this decision				x	x	Rotation rate error (fast or slow)	TLM data	Left display	0:00:15		controller is following a decision tree in the Pass Plan
1.3.81		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.82		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.83		Click on OK box to respond to the restricted command box			x					mouse, left display	0:00:05		This box is an alert to remind the MC that this command can have a large impact on the well being of the satellite. The MC must be sure that the command is correct
1.3.84		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.85		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.86	-	Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.87		Verify that the command was sent and the number of times it was sent				x	x	Number of transmissions intended and number performed	The number of transmissions performed is on the Pass Plan display. The number intended is from the computations performed by the controller and SE	Left display	0:00:15	Numbers displayed on the Pass Plan page adjacent to the step	
1.3.88		On the ACS DELTA V MANEUVER page (right display) verify that the S100 command appears the number of command repititions you entered				x	x	Number of times the command should have been repeated and the number of times the command was actually repeated	TLM data	Right display	0:00:15	This command will appear on the list a number of times. If the command was sent ten (10) times, then S100 will be listed 10 times.	

TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MODALITY MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.89		Go back to the pass plan and read instructions				x				Left display	0:00:10		
1.3.90		Verify DSPIN value on ACS DELTA V MANEUVER page. If not close enough to zero to suit SE go back to pass plan and execute the "Second volley" procedure. If spin is OK, proceed				x	x	Spin rate error (DSPN), should be zero	TLM page	Right display	0:00:15		Inspect the data on the ACS DELTA V MANEUVER page. If it is too far from desired, then do a second maneuver to bring the value within limits
1.3.91		If you are doing a second volley (a second set of commands) recompute the number of times to send command			x		x	Spin rate error, plenum pressure	TLM data	Pencil & paper or calculator	0:01:00		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.92		Click on configure commanding button			x					mouse, left display	0:00:05		
1.3.93		Click on Set Mode Parameters button in the Contact Configuration window to bring up Repeat Mode Parameters box			x					mouse, left display	0:00:05	The Repeat Mode Parameters pop up window appears	
1.3.94		Click on the box with the number of times to repeat the command			x					mouse, left display	0:00:05	Window is highlighted	
1.3.95		Type in the number of times you want the command to execute follwed by <cr>			x					kbd	0:00:15	Number appears in the box as it is typed in	
1.3.96		Press the ENTER key			x					kbd	0:00:05	Highlighting is removed from the window	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.97		Verify that number was entered				x	x	Intended value and value entered	Intended value from controller & SE computations, value entered is on the pass plan screen	Left display	0:00:05		
1.3.98		Click DONE button			x					mouse, left display	0:00:05	Pop Up window is removed from the display	
1.3.99		Verify number of repetitons in the window				x	x	Intended value and value entered	Intended value from controller & SE computations, value entered is on the pass plan screen	Left display	0:00:10		This is the total number of times the command will have been sent when the latest batch is executed. It might be better if this was the number of times the command will be sent on this iteration, rather than the

TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MODALITY MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													total
1.3.100		Click the Close Window box			x					mouse, left display	0:00:05	The pop up window is removed from the display	This is a small box with an "X" in it. The box is located in the upper left hand portion of the pop up window.
1.3.101		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.102		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.103		Click ENABLE to authorize sending restricted command			x					mouse, left display	0:00:05	Confirms that controller knows a restricted command is being sent	
1.3.104		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.105		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.106		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.107		Go to ACS (right display) and verify DSPIN and other values are acceptable				x	x	Spin rate error	TLM data	mouse, right display	0:00:15	Read the number from the ACS DELTA V MANEUVER page	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.108		Read the instructions on the Pass Plan				x				Left display	0:00:10		
1.3.109		Click on Configure the Repeat Command button			x					mouse, left display	0:00:05	A pop up window is displayed on the left monitor	
1.3.110		Click on NORMAL MODE button			x					mouse, left display	0:00:05	The circle in front of the selected option becomes filled in. (The circles are hollow when the option is not selected)	This could probably be automated. When the controller moves on, this should be restored to the safe, default state without any additional controller action.
1.3.111		Click the x box to close the window			x					mouse, left display	0:00:05		
1.3.112		Click on down arrow to continue pass plan			x					mouse, left display	0:00:05	The next page of the Pass Plan appears	
1.3.113		Read the instructions in the Pass Plan				x				Left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.114		Verify that the SPINA value on the ACS DELTA V MANEUVER page matches the value SE entered into the pass plan				x	x	Desired value and actual values of SPINA	Desired value is in the pass plan, actual value in the ACS DELTA V MANEUVER page	Right display	0:00:15		
1.3.115		Verify DSPIN is in the range of plus or minus 0.4				x	x	DSPIN value	DSPIN value is in the ACS DELTA V MANEUVER page	Right display	0:00:15		after first volley you would continue from page 5 to page 6 in the pass plan. Page 6 contains information about the second and subsequent volleys
1.3.116		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.117		Click on the "MAKE CURRENT" item in the pop up			x					mouse, left display	0:00:05	An arrow appears to the left of the command	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		window											
1.3.118		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.119		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.120		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.121		Verify CMD REG value on ACS DELTA V MANEUVER screen				x	x	Desired and actual values	Desired value is in the pass plan, the actual value is in the TLM	Right display	0:00:15		
1.3.122		Click on the command in the left			x					mouse, left display	0:00:05	A pop up window appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		screen.											
1.3.123		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.124		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.125		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.126		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.127		Verify two more values (earth pointing error and SUBMOD) on the ACS DELTA V MANEUVER screen. EPE will be scrolling				x	x	Desired and actual values of these variables	Desired values are in the pass plan, actual values are in the TLM data	Right display	0:00:20		These values are close together on the page, so it takes only slightly longer to verify both values than if just one value was being verified
1.3.128		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.129		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.130		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.131		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
												satellite is communicating	
1.3.132		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.133		Verify CMDREG value on the ACS DELTA V MANEUVER page				x	x	Desired and actual values	Desired value is in the pass plan, actual value is in the TLM data	Right display	0:00:15		Value in Pass Plan
1.3.134		Click on the down arrow on the Pass Plan to go to next page			x					mouse, left display	0:00:05	Next page of the Pass Plan appears	
1.3.135		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.136		Click on the "MAKE CURRENT" item in			x					mouse, left display	0:00:05	An arrow appears to the left of the command	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		the pop up window											
1.3.137		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.138		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.139		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.140		Verify INHIBIT value on ACS DELTA V MANEUVER page				x	x	Desired and actual values	Desired value is in the pass plan, actual value is in the TLM data	Right display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.141		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.142		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.143		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.144		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.145		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.146		Verify ASJEBB value on ACS DELTA V MANEUVER page				x	x	Desired and actual values	Desired value is in the pass plan, actual value is in the TLM data	Right display	0:00:15		
1.3.147		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.148		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.149		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.150		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.151		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.152		Verify CMDREG value on the ACS DELTA V MANEUVER page				x	x	Desired and actual values	Desired value is in the pass plan, actual value is in the TLM data	Right display	0:00:15	Arrow moves showing progress executing procedure	
1.3.153		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.154		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.155		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.156		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.157		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.158		Verify DEL value is what SE specified on the ACS DELTA V MANEUVER page				x	x	Desired and actual values	Desired value is in the pass plan, actual value is in the TLM data	Right display	0:00:15		
1.3.159		Click on the down arrow in the pass plan to continue			x					mouse, left display	0:00:05	Next page of the Pass Plan appears	
1.3.160		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.161		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.162		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	Arrow turns pink	
1.3.163		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.3.164		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.165		Verify CMDREG value on the ACS DELTA V MANEUVER page				x	x	Desired and actual values	Desired value is in the pass plan, actual value is in the TLM data	Right display	0:00:15		
1.3.166		Click on the command in the left screen.			x					mouse, left display	0:00:05	A pop up window appears	
1.3.167		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.3.168		Click on the arrow to the left of the command to make it active			x					mouse	0:00:05	Arrow turns pink	
1.3.169		Verify that data is being received at the once per second rate				x	x	Controller determines if satellite is receiving/transmitting information	display of once per second data	Left display	0:00:05	data is observed to be changing at about once per second, indicating that the satellite is communicating	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.170		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	Arrow turns blue	
1.3.171		Verify AGGAEB value on the ACS DELTA V MANEUVER page				x	x	Desired and actual values	Desired value is in the pass plan, actual value is in the TLM data	Right display	0:00:15		
1.3.172		Go back to pass plan. Pass plan instructions another 20 minute wait for another MOMEST.				x				Clock	0:20:00		
1.3.173		On right screen click on LINK1 DISPLAY button			x					mouse, right display	0:00:05		
1.3.174		Click on MOMEST button			x					mouse, right display	0:00:05		
1.3.175		Watch RESIDUAL MOMENTUM VALUE				x				Right display			this can take another 20 minutes for the

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		converge											values to converge
1.3.176		When converged print the MOMENST results by clicking on the wall paper			x					mouse, right display	0:00:05		
1.3.177		Click on PRINT WINDOW			x					mouse, right display	0:00:05		
1.3.178		put cross hair in window and click to print the window			x					mouse, right display	0:00:05		
1.3.179		Go back to the pass plan and click the end of procedure (endpost button)			x					mouse, left display	0:00:05		
1.3.180		Click the endpost button again. This brings up the SOH pass plan			x					mouse, left display	0:00:05		
	End of MCMEST/ Spin Control procedure												

APPENDIX 5 - IMPACT SENSOR – IRON 3160

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1	Begin Health+ Tracking (see 3160 Pass Procedurs)												System setup is complete and the ARTS is tracking the satellite, data is coming from the satellite and commands are being sent from the OpsCen
1.1.1		Verify SYNC blob is green on left monitor (The "synch blob" is located in the lower right quarter of the screen)				x	x	SYNC status	SYNC indicator in the lower right portion of the left screen	left display	0:00:15	Green blob displayed if SYNC exists, red if not SYNCHED	If the circle (blob) is not filled with green, then you are not communicating with the satellite. The book tells you what contingency plan to run if it is not green
1.1.2		Verify TLM data is being displayed on SNAP screen on				x	x	Controller verifying TLM being updated	TLM data screen	SNAP screen on the right display	0:00:15	Values are non zero and changing	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		right workstation											
1.1.3		Click on the desired button to select frame displayed on the right screen. The button is one of a set in the lower left hand corner of the display.			x					mouse, right display	0:00:05	Button highlights and screen changes	The information to be looked at is specified either in the pass plan or by the Satellite Engineers if it is out of the ordinary.
1.2	Get Vehicle status binder												The binder contains the information recorded during the last pass
1.2.1		Fill in worksheet in binder (julian date, time, name. Command vehicle counts,)			x					grease pencil/water based felt pen	0:00:30		The old entrys are erased and replaced with the current values

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.2		Verify that the current configuration of the satellite matches what is in status binder (128K or 1K data)(antenna type [high gain or low gain antenna, or the omni antenna)				x	x	determine if satellite state matches expected state	Binder contains expectd state, TLM contains actual state	Right display screen and the correct page from the notebook containing the values that are expected.	0:00:15		Compare values in book with values on the screen. These data are on the lefts side of the SNAP screen on the right display.
1.2.3		Verify Battery configuration (first letters of battery acronyms [e.g., TDMRM] on right workstation)				x	x	Current and expected states	Binder contains the expected state, the TLM contains the actual state	SNAP screen on the right display, lower center column contains battery data	0:00:15		The acronyms are not intuitive. (T = trickel, D = disconnect, M = manual, R = recondition, M = Manual)
1.2.4		Click on the button to switch to Bus SOH page on the right display			x					mouse, right display	0:00:05		This screen is used to verify unit processing TLM

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.5		Verify DTU = A is OFF				x	x	Current and expected states	Expected state is in the pass plan, the actual state is in the TLM data	Bus SHO page on th eright display	0:00:15	OFF printed in red	The DTU field is in the lower portion of the second (of four) data columns from the right side of the Bus SOH page.
1.2.6		Verify DTU = B is ON				x	x	Current and expected states		Bus SHO page on th eright display	0:00:15	ON printed in green	Find the BTU field and insure the value is B
1.2.7		Verify Link 1 status (EKG1aB)				x	x	Current and expected states		Bus SHO page on th eright display	0:00:15	red is ON, green is OFF	Find the EKG1aB field. This is the topmost field in the fourth data column from the left on the BUS SOH page
1.2.8		Click on the button to go back to SNAP frame			x					mouse, right display	0:00:05		
1.2.9		Verify mnemonic inhibit = 8				x	x			right display	0:00:15		Compare value in field with desired value. The value is the upper most

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													value in the fourth data colun from the left on the SNAP page.
1.2.10		Verify TCS (theremal control system) status on SNAP screen (about 7000 RPM indicates coolant circulating)				x	x			right display	0:00:15		If the RPM is low, then coolant is not circulating . If the RPM reads about 7000 then the pump is circulating coolant. The value is in the lower part of the first data column from the left on the SNAP page.
1.2.11		Determine the next step by reading the flow chart on the left display				x				Left display	0:00:10		
1.2.12		Verify GAIN HI or GAIN LOW on SNAP				x	x			right display	0:00:15		Visual inspection . This must match

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		page											tabled value. The value is the uppermost value in the left portion of the SNAP screen.
1.2.13		Verify VDBand (veriable dead band) value (0.034 or 0.228 are values on this satellite)				x	x			right display	0:00:15		Visual inspection. This must match tabled value. The value is in the upper portion of the leftmost data column on the SNAP page. (This is pointing error of the satellite.)
1.2.14		Determine the next step by reading the flow chart on the left display				x				Left display	0:00:10		
1.2.15		Verify EPER (earth pointing error) on				x	x			right display	0:00:15		Visual inspection. This must match

TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MODALITY MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		SNAP page											tabled value. This is below the variale error value in the upper part of the leftmost column of the SNAP page
1.2.16		Determine whether or not Plenum has been pumped to make thursters work. Has to be pumped every 8 hrs or so. This is a check to see if pressures are within limits				x	x			Right display, SNAP screen	0:00:15		Examine time of last plenum pressure check and compare with current time. If not within limits pumping is needed to bring pressure to within bounds
	Conclusion o' configuration verfication												
1.3	Begin SOH												
1.3.1		Start with SNAP page								Right display			

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.2		Compare values with values in book, item by item. There are about 61 items to be checked.				x	x			right display, satellite binder	0:00:15		MC goes item by item comparing nominal values/ranges with those in book. If outside of bounds then MC will record the out-of-tolearance values and advise/brief the SE. Tabled values are in binder.
1.3.3		ISAALA is impact sensor, will be YES if impact exceeds limit, NO if no impacts greater than or equal to limit. (Can be set off by solar wind!) For the next step in this task analysis it is assumme					x				0:00:15		On 3160 the impact sensors are set to be triggered at 10G. Other satellites have different values. If this flag is set it means that at least one impact was detected. It doesn't tell you how many

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		d that the Impact Sensor has been triggered and the MC is to reset it.											impacts have been detected.
	Concludes State of health. MC would complete SOH even if an anomaly were found.												
1.4	Begin ImpactSensor reset												
1.4.1		Go back to left hand screen, Click on CONTACT			x					mouse, left display	0:00:05		CONTCT button is an item on the menu at the top of the screen. It causes a pull down menu to appear

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.4.2		Click on EXECUTE COMMAND PROCEDURE in the pull down menu			x					mouse, left display	0:00:05	a window opens and the pull down menu disappears when the MC clicks on this item	
1.4.3		Click on the arrow key to bring up the list of available procedures. (a pull down menu)			x					mouse, left display	0:00:05		The arrow (a triangle) brings up a "pick list". The MC can select the next command procedure from this "Pick List"
1.4.4		Click on the Select Command procedure desired in the list of available procedures			x	x				mouse, left display	0:00:05	A second level "pick list" containing the commands in that group appears	
1.4.5		Click on the procedure that should be executed in the list of procedures			x					mouse, left display	0:00:05	Selected item is highlighted and entered into the text box.	The MC must either memorize the groupings of the functions, or must manually search the lists.

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.4.6		Verify that correct pass plan selected.				x	x			left display	0:00:15		The pass plans selected are from the Pass Action Plan
1.4.7		Click OK			x					mouse, left display	0:00:05		
1.4.8		Read Pass Plan to refresh MC on what Pass Plan does.				x				left display	0:00:10		The text tells the MC what the pass plan does, and something about the actions that should be taken. This can include criteria for contacting a SE.
1.4.9		Click on the SNAP page button in the right screen			x					mouse, right display	0:00:05		
1.4.10		Go back to left screen and continue reading pass plans				x				lef tdisplay	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.4.11		Go back to right screen (SNAP Page) and visually verify ISALLA and/or ISAALB = NO				x	x			Right display	0:00:20		Values are located in the upper portion of the second data column from the left on the SNAP screen.
1.4.12		Go back to left screen. Rerun SOH per pass plan. This will insure that nothing else is out of limits.			x	x	x			Both screens and the list of nominal values.			This process is outlined above. The purpose of redoing the SOH is to determine if the impact effected other systems.
1.4.13		If something else is out of limits you right it down. Complete SOH so you have enough information before going and getting SE			x	x				Paper and pencil, Right display. Or a screen dump of the right display to get record of values	0:00:30		If the values of any parameters are out of limts record them for use in briefing SE. If no values out of limits then reset the flag.

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.4.14		Go to next page on Pass Plan by clicking on the down arrow			x					mouse, left screen	0:00:05		
1.1.15		Read the next page of the pass plan				x				left display	0:00:10		
1.1.16		Go to SNAP screen on right display and determine if A side or B side (ISAALA or ISAALB) is YES				x	x			right display	0:00:15		Visual inspection. You want to determine if the a or B side detector was triggered. You need this to follow the decision tree in the pass plan.
1.1.17		For the side you want to change, left click on command desired in the pass plan			x					mouse, left display	0:00:15		on the pass plan on the left display

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.4.18		Click on MAKE CURRENT from the pull down menu			x					mouse, left display	0:00:05	When the command is "Made Current", an arrow appears to the left of the command in the Pass Plan	on the pass plan on the left display
1.4.19		Click on the ARROW to execute the current command It turns pink to let you know that you are about to execute the command			x					mouse, left display	0:00:05	button turns pink when ready to be executed. (Actually, the first click loads the command onto the workstation and the second click executes the command.)	on the pass plan on the left display
1.4.20		Verify that data is being received at the once per second rate.				x	x	Controller determines if satellite is receiveing /transmittiong data	Display of once per second data	Left display	0:00:05	Data is observed to be changing at about once per second, indicating that the satellite is communicating	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.4.21		Click on the arrow again to execute that command			x					mouse, left display	0:00:05	Button next to the step will turn green when command sent, red if command not sent	on the pass plan on the left display
1.4.22		Read the value on the SNAP page (right hand screen)				x				Right - display	0:00:10	The value of the impact sensor will change. The MC will repeat the comand until this value reaches the desired value.	
1.4.23		The next step in the pass plan tells MC to repeat previous command until ISAALA = NO and ISASTA = 10G. This may require a number (e.g., eight) replications of the command				x	x			ISAALA flag and ISASTA value on the right display contain the criteria. The left display is where the MC initiates the action to change these values		read value on the right display. If the flag is not set or the value <> 10 G then repeat the process.	This time varies epending on the number of cycles required

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		to get to the desired value. It is a circular command.											
1.4.24		Make END POST step current (end post is a semi-circle with an up triangle in it.)			x					mouse, left display	0:00:05		The pass plan contains an estimate of how long this procedure should take. This information is just before the end post. It really should be at the start so that the MC can determine if enough time is available in the contact to complete the procedure and to accomplis

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													h the other goals for that contact.
1.4.25		Click on end post will send you to pass plan.			x					mouse, left display	0:00:05	The left display returns to the pass plan	
1.4.26		Pass through steps in pass plan that you don't want to execute			x					mouse, left display	0:00:45		
1.4.27		Write T&V count in satellite configuration status sheet			x					grease pencil/water based marker in the satellite's configuration log book	0:00:30		
	Finished Impact Sensor												
1.5	Return to SOH												
1.5.1	-	Click on the SAVE CURRENT CONFIG command			x					mouse, left display	0:00:05	A pop up window appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.5.2		Click on the "MAKE CURRENT COMMAND ACTIVE" item in the pop up window			x					mouse, left display	0:00:05	arrow appears in the pass plan	
1.5.3		Click on the arrow to the left of the command to make it active								mouse. Left display	0:00:05	Arrow turns pink	
1.5.4		Click on the arrow to the left of the SAVE CURRENT CONFIG command to execute the command			x					mouse. Left display	0:00:05	Arrow turns blue.	
1.6	Fade the support												"Fading" means to end the contact
1.6.1		Click GO PASSIVE midway down the screen on the right side of the screen (outside the pass plan area)			x					mouse, left display	0:00:05		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.6.2		Click APA RESET midway down the screen on the right side outside of the pass plan area			x					mouse, left display	0:00:05		
1.6.3		Click DISCONNECT FROM RTS This button is loated near the top of the screen on the right side, outside the pass plan area			x					mouse, left display	0:00:05		
1.6.4		Click DECONFIGURE AIM. This button is located near the bottom on the right side of the screen outside of the pass plan area			x					mouse, left display	0:00:05		
1.7	Continue with steps in Pass Plan to end the contact												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.7.1		Click STORE CONFIG TO SYBASE in the pass plan			X					mouse, left display	0:00:05		
1.7.2		Click on the GO AUTOMATIC step on the left screen			x					Mouse, left display	0:00:05	A pop up window appeard	
1.7.3		Click on the "MAKE CURRENT" item in the pop up window									0:00:05	An arrow appears to the left of the command	
1.7.4		Click on the arrow to the left of the command to make it active								Mouse, left display	0:00:05	The arrow turns pink	
1.7.5		Click on the arrow a second time to execute the command			x					Mouse, left display	0:00:05	The arrow turns blue	workstation steps through about five steps in the pass plan automatically. Some steps list their progress in a window located in the upper

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													center portion of the screen. The MC watches this window to see if progress is occuring at the expected rate.
1.7.6		Click on the GO MANUAL command			x	x				Mouse, left display	0:00:05	GO MANUAL step appears on the left display. Also, the TLM blob should be read, indicating that you are no longer getting telemetry.	
1.7.7		Click on the "MAKE CURRENT" item in the pop up window									0:00:05	An arrow appears to the left of the command	
1.7.8		Click on the arrow to the left of the command to make it								Mouse, left display	0:00:05	The arrow turns pink	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		active											
1.7.9		Click on the arrow a second time to execute the command			x					Mouse, left display	0:00:05	The arrow turns blue	
1.7.10		Click on the endpost to go back to the main pass plan. This will take you to an end post.			x					Mouse, left display	0:00:05	Original State of Health (SOH) pass pla nreappears on the screen	
1.7.11		On the main pass plan click the endpost to end the contact.			x					Mouse, left display	0:00:05	Pass Plan area is blank showing that the pass plan has been completed	

APPENDIX 6 - SET LINK 2 TO 128K HIGH PLUS SET THERMAL CONTROL SYSTEM (TCS) TO CIRCULATE – IRON 3160

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1	Begin LINK-2-to-128K-Hi procedure. You are at the point in the SOH procedure where you are prompted to execute any additional pass plans you want on this support session.												
1.1.1		Click on CONTACT PROCEDURES on left screen			x					mouse, left display	0:00:05	Pop up window appears	
1.1.2		Click on Execute Command Procedures in the pop up window			x					mouse, left display	0:00:05		
1.1.3		Click on the arrow next to the Pass Plan name field in the pop up window			x					mouse, left display	0:00:05	A repository of command procedure groups appears	
1.1.4		Click on the desired			x					mouse, left display	0:00:05	a list of the pass plans in	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		group of pass plans										that group pops up	
1.1.5		Scroll through the list of pass plans to find desired pass plan			x					mouse, left display	0:00:05		
1.1.6		Click on the desired pass plan			x					mouse, left display	0:00:05	The list of pass plans is remved from the screen and the desired pass plan in entered into the name field	
1.1.7		Click on OK			x					mouse, left display	0:00:05	The Pass Plan selection window is removed from the screen, and the new pass plan is displayed in the left screen	
1.1.8		The pass plan now appears on the left display (aka. the G2 window). Read the				x				Left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		instructions											
1.1.9		Click on the LINK 2 DISPLAYS button on the right screen			x					mouse, right display	0:00:05	The LINK 2 page is displayed on the right screen. This screen has a set of buttons	
1.1.10		Click on the Comm System Config button on the LINK 2 page			x					mouse, right display	0:00:05	A page of Link 2 TLM data is displayed on the right monitor	
1.1.11		Read the instruction in the pass plan				x				Left display	0:00:10		
1.1.12		Verify Et2AOB = On on the right screen				x	x			Right display (to see the actual value in the TLM data)	0:00:15		
1.1.13		Read the instruction in the pass plan				x				Left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.14		Verify EDBBRD = the current data rate (1K would appear)				x	x			Left display (to see what variable to read and the expected value) and the right display (to see the actual value in the TLM data)	0:00:15		
1.1.15		Read the next step from the pass plan				x				Left display	0:00:10		
1.1.16		Click on BUS SOH button on the right screen			x					mouse, right display	0:00:05	data screen appears	
1.1.17		Go back to the left screen and read instruction. This tells you that the action is to set the hi gain antenna ON				x				Left display	0:00:10		
1.1.18		Click on the command in the Pass Plan			x					mouse, left display	0:00:05	a pop up menu appears	
1.1.19		Click on MAKE CURREN			x					mouse, left display	0:00:05	an arrow appears to the left	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		T in the pop up menu										of the command	
1.1.20		Click on the arrow to the left of the command			x					mouse, left display	0:00:05	the arrow turns pink	
1.1.21		Verify Once Per Second data				x				left display	0:00:05	Data updating	
1.1.22		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	the arrow turns blue	
1.1.23		Read the next instruction in the pass plan				x				Left display	0:00:10		
1.1.24		Verify four values on the Bus SOH page. The MC goes back and forth between the screens to check the values				x	x			Left display (to see what variable to read and the expected value) and the right display (to see the actual value in the TLM data)	0:00:30		
1.1.25		Get on the phone and notify the ARTS operator to standby	x	x						Phone	0:00:10	ARTS operator confirms readiness	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		for an increase in signal strength											
1.1.26		Click on the Continue Arrow to go to next page on the Pass Plan			x					mouse, left display	0:00:05	The next page of the Pass Plan appears on the left screen	
1.1.27		Click on the command in the Pass Plan			x					mouse, left display	0:00:05	a pop up menu appears	
1.1.28		Click on MAKE CURRENT in the pop up menu			x					mouse, left display	0:00:05	an arrow appears to the left of the command	
1.1.29		Click on the arrow to the left of the command			x					mouse, left display	0:00:05	the arrow turns pink	
1.1.30		Verify Once Per Second data				x				left display	0:00:05	Data updating	
1.1.31		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	the arrow turns blue	
1.1.32		Read the next instruction in the pass plan				x				Left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.33		Verify four values to insure that command had the intended effect on the BUS SOH screen				x	x			Left display (to identify the variables and the nominal values), and the right display (to see the actual value in the TLM data). The right screen is showing the BUS SOH page of TLM data.	0:00:30		MC goes back and forth between the screens to check the values
1.1.34		Go back to the pass plan and read the instructions.				x				Left display	0:00:10		
1.1.35		Go back to right screen and press the LINK 2 DISPLAY button			x	x				mouse, right display	0:00:05	The LINK 2 DISPLAY page will be displayed.	
1.1.36		Click on COMM SYSTEM CONFIGURATION button on the right			x					mouse, right display	0:00:05	The Comm System Configuration page will be shown on	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		screen										the right display	
1.1.37		Go back to pass plan and read instructions				x				Left display	0:00:10		
1.1.38		Get on phone and alert the ARTS operator that you will be commanding to 128 K data rate.	x	x						Phone	0:00:10	Confirmation from the ARTS operator	
1.1.39		Click command to make 128 K active			x					mouse, left display	0:00:05	A pop up window appears	
1.1.40		Click on MAKE CURRENT in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.1.41		Click on the arrow to the left of the command			x					mouse, left display	0:00:05	The arrow turns pink	
1.1.42		Verify that data is being received at the once per second rate				x				left display	0:00:05	Data is observed to be changing at about once per second, indicating	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
												that the satellite is communicating	
1.1.43		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	The arrow turns blue	
1.1.44		Read instructions in the Pass Plan. This instruction tells the controller to verify a particular value				x				Left display	0:00:10		
1.1.45		Verify value on the COMM SYTEM CONFIGURATION page on the right display				x	x			Right display	0:00:15		
1.1.46		Click on the command in the Pass Plan			x					mouse, left display	0:00:05	a pop up menu appears	
1.1.47		Click on MAKE CURRENT in the pop up menu			x					mouse, left display	0:00:05	an arrow appears to the left of the command	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.48		Click on the arrow to the left of the command			x					mouse, left display	0:00:05	the arrow turns pink	
1.1.49		Verify that data is being received at the once per second rate				x				left display	0:00:05	Data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.1.50		Click on the arrow to the left of the command a second time			x					mouse, left display	0:00:05	the arrow turns blue	
1.1.51		Verify value on the COMM SYSTEM CONFIGURATTION page				x	x			Right display	0:00:15		
1.1.52		Phone ARTS and tell them you have sent the command and ask them to change their patches. You will lose TLM	x	x						Phone	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		until the DCIS and FC puts in patches (30 sec to 1 min)											
	End												
1.2	Begin TCS to Circulate												
1.2.1		Click on CONTACT on left screen			x					mouse, left display	0:00:05	pop up window appears	
1.2.1		Click on Execute Command Procedures			x					mouse, left display	0:00:05	A window where the name of the pass plan to be performed is specified appears on the left display	
1.2.3		Click on the arrow next to the field containing the name of the Pass Plan			x					mouse, left display	0:00:05	A list of the names of the groups of pass plans appears	
1.2.4		Click on the name of the group containing the desired pass plan			x					mouse, left display	0:00:05	A list of pass plans is displayed.	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.5		Click on the desired pass plan			x					mouse, left display	0:00:05	The name of the pass plan will be entered into the proper field, and the windows containing the group names and the names of the pass plan in the group will be removed from the screen.	
1.2.6		Click on OK			x					mouse, left display	0:00:05	The desired pass plan will appear in the left display	
1.2.7		Read the instructions in the pass plan				x				Left display	0:00:10		
1.2.8		Click on the SNAP button on the right workstation			x					mouse, right display	0:00:05	Bring up a different TLM display in the right display	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.9		On SNAP window veritfy state of TCS RPM (about 0 when not pumping, about 7000 rpm if pumping)				x	x			Left display (to identify the variables and the nominal values), and the right display (to see the actual value in the TLM data)	0:00:15		
1.2.10		Read the next instruction in the pass plan				x				Left display (to identify the variables and the nominal values)	0:00:10		
1.2.11		Verify TCaSTA values are about the same as those displayed in the pass plan				x	x			Right display (to see the actual value in the TLM data)	0:00:15		a range =/- 0.4 was in the pass plan
1.2.12		Click on the end arrow to return to SOH pass plan			x					mouse, left display	0:00:05	Returns to the SOH pass plan	
	End of TCS to Circulate Pass Plan												

APPENDIX 7 - ECLIPSE MONITOR PLUS ATTITUDE DATA COLLECTION – IRON 9445

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1	Begin Eclipse Monitor												
1.1.1		Click on CONTACT on the menu bar on the left display			x					mouse, left display	0:00:05	Pull down window appears	
1.1.2		Click on EXECUTE COMMAND PROCEDURE item in the pull down window			x					mouse, left display	0:00:05	a pop up window will appear. This pop up window is where the controller enters the name of the pass plan to be run	
1.1.3		Click on the arrow next to the name field in the pop up window to get respository of procedures			x					mouse, left display	0:00:05	A list of pass plan groups appears	The controller has to know which group contains the task plan that is needed. If the controller doesn't know which group contains the task plan then he must search for it

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													manually; there aren't any aids to finding it.
1.1.4		Click on the BATTERIES-AUTO group			x					mouse, left display	0:00:05	A window containing the task plans in that group opens	
1.1.5		Click on the ECLIPSE MONITOR task plan from the list			x					mouse, left display	0:00:05	The windows containing the list of task plans and task plan groups are remved from the screen and the name of the pass plan is put into the name field in the Selected Task Plan windw	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.6		Click OK			x					mouse, left display	0:00:05	The desired task plan appears in the task plan window on the left monitor, replacing any previous task plan.	
1.1.7		Read note about current being negative during eclipse and positive in sunight on pass plan				x				left console	0:00:10		
1.1.8		Click on the Power Subsystem button on to right display to bring up the desired data page. The button is in the lower left corner of the screen with lots of other buttons.			x					mouse, right display	0:00:05		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.9		Read current and voltage values on Power Subsystem page. The battery current values are in the upper right corner of the data area				x				Right display	0:00:15		
1.1.10		Click on GO AUTOMATIC to make it ready to execute. This is done by clicking on the command in the pass plan on the left monitor			x					mouse, left display	0:00:05		
1.1.11		Click to execute GO AUTOMATIC			x					mouse, left display	0:00:05		
1.1.12		Monitor substeps in the pass plan. If there is a failure,				x				left display	0:01:00		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		determine cause											
1.1.13		Return to the pass plan on the left display and read the next step. (Which tells the controller to verify the main bus voltage)				x				Left display	0:00:10		
1.1.14		Verify BUSLIM (bus limit) is low during eclipse on the Power Subsystem page. These data are located in the upper left corner of the data area on th right display				x				Right Display	0:00:15		
1.1.15		Verify MNBSV (Main Bus Voltage). Should be about 31.6 v during				x				Right Display	0:00:15		Eclipse page is an alternative to the Power Subsystem page

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		support											
1.1.16		Return to the pass plan on the left display and read the next step. (Which tells the controller to verify Battery 1, 2, & 3 settings)				x				Left display	0:00:10		
1.1.17		Verify B1STAT = AHI or ALOW on power subsystem page				x				Right Display	0:00:15		ECLIPSE page gives time history of voltage and amperage values during the pass
1.1.18		Verify B2STAT = AHI or ALOW on power subsystem page				x				right display	0:00:15		
1.1.19		Verify B3STAT = AHI or ALOW on power subsystem page				x				right display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.20		Read the next step in the pass plan. This step reminds the controller to print the Eclipse Page.				x				left display	0:00:10	A pull down menu appears on the right display	
1.1.21		Click on the ECLIPSE page button. This button is in the lower left corner of the right display outside the data area.			x					mouse, right display	0:00:05	the ECLIPSE page appears on the right display	
1.1.22		Left click on the "wall paper" outside the ECLISPE Page			x					mouse, right display	0:00:05	A pop up window appears on the right display	
1.1.23		On the drop down window on the right display click on PRINT WINDOW			x					mouse, right display	0:00:05	a sub window appears to the side of the parent window	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.24		On pull down menu click on REVERSE			x					mouse, right display	0:00:05	the windows disappear.	selecting reverse makes the background on the paper copy white, rather than black as is the screen
1.1.24		Put the cross hair cursor into the desired window and click			x					mouse, right display	0:00:05	When click is accepted a tone is generated and the cursor returns to being an arrow. If the printer is nearby the controller might also hearit begin to print the screen	
1.1.25		Go back to pass plan and read note. If the satellite has not come out of eclipse (i.e., no positive current) then notify SE				x				left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.26		Click on end of pass plan arrow to end eclipse pass plan			x					mouse, left display	0:00:05	The left display returns to the previous pass plan.	
1.2	Begin Attitude Data collect pass plan												
1.2.1		Click on CONTACT on the left display. This is located in the menu bar at the top of the screen			x					mouse, left display	0:00:05	a drop down window appears	
1.2.2		Click on EXECUTE COMMAND PROCEDURE			x					mouse, left display	0:00:05	A pop up window appears. This window is where the name of the pass plan to be executed is entered	
1.2.3		Click on the arrow to get respository			x					mouse, left display	0:00:05	a list of pass plan groups appears.	
1.2.4		Click on 9445-PASS PLANS			x					mouse, left display	0:00:05	A list of the pass plans in this group appears on the	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
												screen	
1.2.5		Click on ATTITUDE-DATA-COLLECT			x					mouse, left display	0:00:05	The lists are removed from the screen and the name of the selected pass plan appears in the field in the pass plan selection window	
1.2.6		Verify command procedure entered into window				x				left display	0:00:10	The name of the correct pass plan is entered into the name field	
1.2.7		Click on OK			x					mouse, left display	0:00:05	The pass plan selection window is removed from the screen	
1.2.8		Begin reading pass plan				x	x			left display	0:00:10		The pass plan won't be visible when these particular tasks are performed.

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													Therefore, the controller either needs to remember the tasks (typical) or write then on scratch paper (very atypical, at least for an experienced controller)
1.2.9		Click on the minimize button (upper right hand corner of the pass plan window on the left display) to downsize Pass Plan Window			x					mouse, left display	0:00:05	pass plan window is minimized	
1.2.10		Locate REAL TIME SESSION EXECUTION Panel				x				left display	0:00:05		This is one of the windows openned early. The Pass Plan window has been over the top of it so it wasn't visible to

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													the controller until the Pass Plan window was minimized
1.2.11		Click in the REAL TIME SESSION EXECUTION PANEL to make that window active			x					mouse, left display	0:00:05	Bar at the top of the window changes color (to purple) to show that it is the active window	
1.2.12		Click on the APPLICATIONS button on the toolbar along the top of the REAL TIME SESSION EXECUTION window			x					mouse, left display	0:00:05	A pull down window appears	
1.2.13		Click on LAUNCH			x					mouse, left display	0:00:05	A large window pops up over the top of the other windows	
1.2.14		Scroll down to find RSC Analysis File Archiever			x					mouse, left display	0:00:10	entries scroll up and down the page	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.15		Click on RSC Analysis File Archiver			x					mouse, left display	0:00:05	the selected text is highlighted	
1.2.16		Click on button to bring up DSP Host options (located in the lower right portion of this window)			x					mouse, left display	0:00:05	a drop down window with allowable options appears	
1.2.17		Click on DSP Host = Workstation 2			x					mouse, left display	0:00:05	the pop up window is removed from the screen	
1.2.18		Click on button to bring up Execution Host options (located in lower left portion of this window)			x					mouse, left display	0:00:05	a pop up window containing the allowable options appears	
1.2.19		Click on the Execution Host = Workstation 2 option			x					mouse, left display	0:00:05	pop up menu is removed from the window	
1.2.20		Click OK (button located in the lower left portion of the			x					moue, left display	0:00:05	this brings up an "Available sublist" panel	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		window)											
1.2.21		Click on Attitude_collect in the Available sublist panel			x					mouse, left display	0:00:05	The selected option is highlighted in the list	
1.2.22		Click on the send button. (located in the lower middle part of the window)			x					mouse, left display	0:00:05	another panel appears	
1.2.23		Type in "ATTITUDE_COLLECT" on the Available Data Groups Panel . The field for this name is in the lower portion of the window.			x					keyboard for left display	0:00:15	Name appears in the field as it is typed in	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.24		Click SEND			x					mouse, left display	0:00:05	The window is removed, revealing another window that had been covered up. In application status panel verify name is on the list, indicating that it launched. If it hangs there will be two asterisks next to it	
1.2.25		Verify that the application was launched by examining the APPLICATIONS STATUS PANEL				x				Left display	0:00:15	Application should be listed. If hung the name will have two stars (**) next to the name	
1.2.26		Click on the Pass Plan icon to maximimize Pass Plan. The window icons are			x					mouse, left display	0:00:05	Pass plan reappears on the left screen	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		located in a column along the left edge of the screen.											
1.2.27		Read pass plan to detemine next action				x				left display	0:00:10	correct indication seen in the data window	
1.2.28		Click minimize button on Pass Plan so that window containing the data to be verified is visible			x					mouse, left display	0:00:05	pass plan window minimized	
1.2.29		Verify application has launched in AWE Message Display window				x	x			left display	0:00:10	read correct data in the window	
1.2.30		Click on Pass Plan icon to restore it to the display			x					mouse, left display	0:00:05	Pass Plan reappears	
1.2.31		Read pass plan for next step (verify connecting of ACS analysis				x				left window	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		window)											
1.2.32		Click on minimize icon on the pass plan window so that other windows containing the information to be verified are visible			x					mouse, left display	0:00:05	pass plan window is minimized	
1.2.33		Click on window containing log of processes called and executed			x					mouse, left display	0:00:05	The header of the window changes color and the window becomes the top window on the screen	
1.2.34		Scroll to find the line showing the status of the particular process.			x	x				mouse, left display	0:00:10		The list moves as the controller scrolls through it. There are no tools that help the controller find the process of interest

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.35		Read the value and make sure it is correct				x	x			left dislpay	0:00:10	read the status of the process and determine if it is correct	
1.2.36		Click on the Pass Plan icon to restore the pass plan			x					mouse, left display	0:00:05	pass plan icon restored on the left screen	
1.2.37		Read the pass plan				x				left display	0:00:05		
1.2.38		Click on the ATTITUDE CONTROL button in the right-hand on the right display to get the desired data on the display			x					mouse, right display	0:00:05	The desired window replaces the window that was displayed.	
1.2.39		Click on wall paper			x					mouse, right display	0:00:05	A drop down window appears	
1.2.40		Click on print window button on the drop down menu			x					mouse, right display	0:00:05	a subwindow appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.41		Click on REVERSE			x					mouse, right display	0:00:05	the drop down window and its sub window are reoved from the screen, and the cursor turns into a cross hair (rather than an arrow)	
1.2.42		Cut cross hairs on window you want to print			x					mouse, right display	0:00:05	Window is highlighted	
1.2.43		Click when cross hair in desired window			x					mouse, right display	0:00:05		
1.2.44		Click on the GENERAL HEALTH button on the right display for printing			x					mouse, right display	0:00:05	The desired window replaces the window that was displayed.	
1.2.45		Click on wall paper			x					mouse, right display	0:00:05	A drop down window appears	
1.2.46		Click on print window			x					mouse, right display	0:00:05	a subwindow appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.47		Click on REVERSE			x					mouse, right display	0:00:05	the drop down window and its sub window are reoved from the screen, and the cursor turns into a cross hair (rather than an arrow)	
1.2.48		Put cross hairs on window you want to print			x					mouse, right display	0:00:05	Window is highlighted	
1.2.49		Click when cross hair in desired window			x					mouse, right display	0:00:05	cursor changes from cross hairs to an arrow	
1.2.50		Read the next step in the pass plan from the left screen				x				left display	0:00:10		
1.2.51		Determine if CTADS = NCW by looking on right display and see if CTADS = North Chord Width				x	x			right display	0:00:15		NCW = north cord width

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		(NCW)											
1.2.52		Use value of CTADS ("NO" in this example) to make the decision described in the Pass Plan					x			left display	0:00:05		The pass plan has a decision point. The controller makes the decision using the value just obtained from the right screen, and follows that path in the pass plan.
1.2.53		Click on the command in the pass plan			x					mouse, left display	0:00:05	A pop up menu appears	
1.2.54		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command, and the pop up menu is removed	
1.2.55		Click on the arrow next to the command			x					mouse, left display	0:00:05	The arrow turns pink	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.56		Verify that data is being received at the once per second rate				x				left display	0:00:05	Data is observed to be changing at about once per second, indicating that the satellite is communicating.	
1.2.57		Click to execute the command			x					mouse, left display	0:00:05	The arrow turns blue	
1.2.58		Verify command was performed				x				Left display	0:00:15	The icon in the pass plan turns green	In the simulation this step failed due to limitations in the simulator
1.2.59		Verify that CTADS value is now correct in the General Health window				x				Right display	0:00:15	The value in the TLM matches the expected value	
1.2.60		Read the next step in the Pass Plan. (It is telling the controller to make screen prints)				x				Left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.61		Click on wall paper			x					mouse, right display	0:00:05	A drop down window appears	
1.2.62		Click on print window			x					mouse, right display	0:00:05	a subwindow appears	
1.2.63		Click on REVERSE			x					mouse, right display	0:00:05	the drop down window and its sub window are reoved from the screen, and the cursor turns into a cross hair (rather than an arrow)	
1.2.64		Put cross hairs on window you want to print			x					mouse, right display	0:00:05	Window is highlighted	
1.2.65		click when cross hair in desired window			x					mouse, right display	0:00:05	cursor changes from cross hairs to an arrow	
1.2.66		Click on the ATTITUDE CONTROL button on the right display to select this			x					mouse, right display	0:00:05	ATTITUDE CONTROL window appears on the right screen	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		window for printing											
1.2.67		Click on wall paper			x					mouse, right display	0:00:05	A drop down window appears	
1.2.68		Click on print window			x					mouse, right display	0:00:05	a subwindow appears	
1.2.69		Click on REVERSE			x					mouse, right display	0:00:05	the drop down window and its sub window are reoved from the screen, and the cursor turns into a cross hair (rather than an arrow)	
1.2.70		Put cross hairs on window you want to print			x					mouse, right display	0:00:05	Window is highlighted	
1.2.71		Click when cross hair in desired window (ATTITUDE CONTROL)			x					mouse, right display	0:00:05	cursor changes from cross hairs to an arrow	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.72		Go back to Pass Plan and read the next step. There is a 5 minute wait before continuing				x				left display, clock or timer for the 5 minute wait	0:00:10		
1.2.72		Click on the ATTITUDE CONTROL button on the right display to select this window for printing			x					mouse, right display	0:00:05	ATTITUDE CONTROL window appears on the right screen	The ATTITUDE CONTROL screen is likely to already be up on the right display, depending on the order that the two screens were printed earlier
1.2.73		Click on wall paper			x					mouse, right display	0:00:05	A drop down window appears	
1.2.74		Click on print window			x					mouse, right display	0:00:05	a subwindow appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.75		Click on REVERSE			x					mouse, right display	0:00:05	the drop down window and its sub window are reoved from the screen, and the cursor turns into a cross hair (rather than an arrow)	
1.2.76		Put cross hairs on window you want to print			x					mouse, right display	0:00:05	Window is highlighted	
1.2.77		Click when cross hair in desired window (ATTITUDE CONTROL)			x					mouse, right display	0:00:05	cursor changes from cross hairs to an arrow	
1.2.78		Click on the continue arrow in front of the next step in the pass plan to continue			x					mouse, left display	0:00:05	the next page of the pass plan appears on the left display	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.79		Click on the first command (Switch CTADS back to SCW [South Chord Width])			x					mouse, left display	0:00:05	A pop up window appears	
1.2.80		Click on the "MAKE CURRENT" item in the pop up window			x					mouse, left display	0:00:05	An arrow appears to the left of the command	
1.2.81		Click on the arrow to make the command active			x					mouse, left display	0:00:05	The arrow turns piink	
1.2.82		Verify that data is being received at once per second				x				Left displayu	0:00:05	Data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.2.83	-	Click on the arrow to execute the command			x					mouse, left display	0:00:05	The arrow turns blue	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.84		Read the next step in the pass plan				x				left display	0:00:10		The SME explained at this point that whenever he execute a command he is making other visual checks to make sure it is progressing normally. Once per second data is one thing that is checked, and the blob that shows telemery status. These are not located together or in a conspicuous location. These are continuously scanned, particularly before sending a command

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.85		Verify CTADS = E-E on right display (Upper left portion of the data area on the right display)				x				right display	0:00:15		
1.2.86		Go back to Pass Plan and read the next step (this says to wait another 5 minutes and then do some more screen printing)				x				left display, timer or clock	0:00:10		
1.2.87		Click on the ATTITUDE CONTROL button on the right display (unless this window is already being displayed) for printing			x					mouse, right display	0:00:05	the window comes up on the right display	The ATTITUDE CONTROL window may already be up on the right screen
1.2.88		Click on wall paper			x					mouse, right display	0:00:05	A drop down window appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.89		Click on print window			x					mouse, right display	0:00:05	a subwindow appears	
1.2.90		Click on REVERSE			x					mouse, right display	0:00:05	the drop down window and its sub window are reoved from the screen, and the cursor turns into a cross hair (rather than an arrow)	
1.2.91		Put cross hairs on window you want to print			x					mouse, right display	0:00:05	Window is highlighted	
1.2.92		Click when cross hair in desired window (ATTITUDE CONTROL)			x					mouse, right display	0:00:05	cursor changes from cross hairs to an arrow	
1.2.93		Click on the command			x					mouse, left display	0:00:05	A pop up window appears	
1.2.94		Click on the "MAKE CURRENT" item in the pop			x					mouse, left display	0:00:05	An arrow appears to the left of the command	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		up window											
1.2.95		Click on the arrow to make the command active			x					mouse, left display	0:00:05	The arrow turns piink	
1.2.96		Verify that data is being received at once per second				x				Left displayu	0:00:15	Data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.2.97		Click on the arrow to execute the command			x					mouse, left display	0:00:05	The arrow turns blue	
1.2.98		Go back to pass plan and read									0:00:10		
1.2.99		Click on command (to set CTADS back to NCW)			x					mouse, left display	0:00:05		
1.2.100		Click on the "MAKE CURRENT" item in the pop up			x					mouse, left display	0:00:05	An arrow appears to the left of the command	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		window											
1.2.101		Click on the arrow to make the command active			x					mouse, left display	0:00:05	The arrow turns piink	
1.2.102		Verify that data is being received at once per second				x				Left displayu	0:00:05	Data is observed to be changing at about once per second, indicating that the satellite is communicating	
1.2.103		Click on the arrow to execute the command			x					mouse, left display	0:00:05	The arrow turns blue	
1.2.104		Verify that CTADS = NCW on the ATTITUDE CONTROL PAGE on the right screen				x				right display	0:00:15		
1.2.105		Read next step on Pass Plan				x				left display	0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.106		Click on the IMT button on the right display for printing			x					mouse, right display	0:00:05	IMT screen appears on the right display replacing the screen that was displayed perviously	
1.2.107		Click on wall paper			x					mouse, right display	0:00:05	A drop down window appears	
1.2.108		Click on print window			x					mouse, right display	0:00:05	a subwindow appears	
1.2.109		Put the cross hairs on window you want to print			x					mouse, right display	0:00:05	the drop down window and its sub window are reoved from the screen, and the cursor turns into a cross hair (rather than an arrow)	
1.2.110		Click when cross hair in desired window (IMT window)			x					mouse, right display	0:00:05	cursor changes from cross hairs to an arrow	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.111		Collect printouts from the printer and put them in the binder for the Ses			x						0:01:00		
1.2.112		Click on the end post icon to go back to 9445 pass plan			x					mouse, left display	0:00:05		
	Erd of Edipse Monitor												

APPENDIX 8 - HEALTH & TRACKING PLUS BATTERY 1 DISCHARGE MONITOR PLUS NO TLM PROCEDURE – IRON 9445

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1	Begin SOH												only three points are checked on this satellite
1.1.1		Click on pass plan to start SOH procedures			x					mouse, left display	0:00:05		
1.1.2		Read the first step in the pass plan				x				Left display	0:00:10		Note the values that are to be examined
1.1.3		Verify values of CMD1CK and CMD2CK				x	x			Right display	0:00:15		Compare the telemetered values with the values in the pass plan. (The pass plan in on the left hand screen at this point.) If out of limits notify SE. There telemetered values are in the upper left portion of the TT&C STATUS page on the right monitor.

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.4		Read the step in the pass plan				x				Left display	0:00:10		Note the values that are to be examined
1.1.5		Verify the values of EIAMAT and SLAMAT									0:00:15		The telemetered values are in the right hand column of the data area on the TT&C page on the right hand monitor. Compare the telemetered values with the nominal values as listed in the pass plan.
1.1.6		Read the next step in the pass plan				x				Left display	0:00:10		
1.1.7		Click on the next step in the pass plan			x					mouse, left display	0:00:05	A pop up window appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.8		Click on the MAKE CURRENT item in the pop up window			x					mouse, left display	0:00:05	The pop up window is removed from the display and an arrow appears to the left of the command	
1.1.9		Click on the arrow to the left of the command to make it active			x					mouse, left display	0:00:05	the arrow turns pink	
1.1.10		Verify that data is being received at once per second				x				Left display	0:00:05		
1.1.11		Click on the arrow a second time to execute the command			x					mouse, left display	0:00:05	the arrow turns blue	
1.1.12		Verify system has checked batt voltages				x	x			right display	0:00:15		If test fails look at right display to get actual values. Record values and alert a SE

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1.13		Read instructions on pass plan.				x				left display	0:00:10		
1.1.14		Click on power frame button to bring up the POWER SUBSYSTEM page on the right screeen			x					mouse, right display	0:00:05	Power Subsystem page appears on the right monitor. This allows the voltages to be read.	
1.1.15		Write the battery values (voltages & currents) in the vehicle's STATUS BINDER when doing the battery recondition procedure			x	x				Vehicle status binder and right display, grease pencil	0:00:30		Batttery voltaes are in the upper right portion of the Power Subsystem page.
1.2	Begin Battery Reconditi on pocedure												
1.2.1		Examine Vehicle Status log				x				Vehicle Status Binder	0:01:00		
1.2.2		Compare loged values with values on				x	x			Right Display and Vechicle Status	0:00:45		compare values on screen with values in

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		the Power Subsystem display on the right screen								Binder			book
1.2.3		Log battery voltages and currents in Vehicle status log for this contact			x					vehicle status log book, grease pencil	0:01:00		log sheet goes to SE. SE does trending
1.3	Execute other pass plans												This is simulating a C5 No Telemetery procedure
1.3.1		Click CONTACT on the left screen (on IMT menu)			x					mouse, left screen	0:00:05	pop up menu appears	
1.3.2		Click on the EXECUTE COMMAND PROCEDURE button from pull down menu			x					mouse, left screen	0:00:05	A window appears. This window has a field for entering the name of the pass plan	
1.3.3		Use down arrow to search for pass plan in respository. This			x					mouse, left screen	0:00:10	a list of procedure groups appears	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		opens pull down menu containing pass plans											
1.3.4		Click on the desired pass plan group			x					mouse, left screen	0:00:05	A list of the pass plans in that group appears in a pop up menu	
1.3.5		Click on "C5-No Telemetry " On a sub-pull down menu			x					mouse, left screen	0:0;5	the repsoitory of pass plans is removed from the display and the name of the selected pass plan appears in the name field	
1.3.5		Read steps				x				left display	0:00:10		MC would rule out that loss of TLM is not a ground system or antenna problem before execting this procedure

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.6		Phone antenna operator to verify that he sees a carrier and a subcarrier. If so, it rules out an ARTS problem	x	x						phone	0:00:10	ARTS operator responds to query.	
1.3.7		Read next step. If at any time you get TLM stop procedure				x				left display	0:00:10		
1.3.8		Click on "Ground Constraint" button on IMT menu			x					mouse, left display	0:00:05	Pull down window appears on the left screen. This window has several non-exclusive options the controller can select.	IMT is the let screen
1.3.9		Click on VCC and GCC Match and TLM Synch Lock to release constraints			x					mouse	0:00:05	Option highlights in the pull down window	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.10		Verify that the constraints are inactive visually				x	x			left display	0:00:10	Status of ground constraint indicator	
1.3.11		Click to close the window			x					mouse, left display	0:00:05		
1.3.12		Record GCC and VCC expected values manually			x					pencil and paper. Record the values on scratch paper for use during the procedure	0:00:30		
1.3.13		Read the procedure . It refers you to the COBRA Troubleshooting binder. You will be running a loss of TLM procedure (step 13) contained in that binder.				x				Left screen	0:00:10		
1.3.14		Click on continue arrow on pass plan			x					mouse, left displau	0:00:05	screen changes when you click on the arrow. It goes to the next step.	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.15		Click on TT&C status on left display			x					mouse, left display	0:00:05	TT&SC Status page appears on the right display	
1.3.16		Verify TT&C count. (If there was never any TLM the TT&C will be blank. You would need to look at the printouts to get last value. If you had TLM at some point during this pass the values on the roght screen would show the last ENCDR values (encoder)).				x					0:00:15		Controller verifys the TLM count value on the right screen.
1.3.17		Determine if the ENCDER =1 or =2.				x				Right display	0:00:15		
1.3.18		Decide which command to execute					x				0:00:10		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		based on the value of ENCDR											
1.3.19		Click on the command selected			x					mouse, left display	0:00:05	A pop up menu appears	
1.3.20		Click on "MAKE CURRENT" in the pop up window			x					mouse, left display	0:00:05	The pop up window is removed from the display and an arrow appears to the left of the command	
1.3.21		Click on the arrow to the left of the command			x					mouse, left display	0:00:05	the arrow turns pink	
1.3.22		Verify once oer second data				x				Left display	0:00:05	Once per second data	
1.3.23		Click on the arrow a second time			x					mouse, left display	0:00:05	The arrow turns blue	
1.3.24		read steps: ("Wait 140 seconds ...)				x				left display	0:02:20		It takes up to 140 sec when the equipment switch is made. The MC must wait and monitor the right display to

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
													see if the change has an effect.
1.3.25		If no TLM in 140 sec alert an SE that there is a problem.				x	x			Right display			
1.3.26		Verify you have TLM on right screen				x				Right display, TT&C Status page	0:00:15	Value in upper portion of he third data column from the left would contain values that increment and the blob on the top of the screen will be green	
1.3.27		Verify you have TLM on left screen				x				Left display	0:00:15	Blob in the lower right portion of the left screen (outside the pass plan area) will be green if TLM is is OK	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.28		Read reminder in the pass plan to log problem into scheduler (scoring pass would cause a failure)				x				Left display	0:00:10		
1.3.29		Click on arrow in the pass plan to continue			x					mouse	0:00:05		
1.3.30		Read next note in the pass plan				x				Left Display	0:00:10		
1.3.31		Verify ENCRP (encrytor) status on the TT&C frame . Normally 1 or 2 or both will be ON				x				Right display	0:00:15		
1.3.32		Click on the next step (S7048) in the pass plan			x					mouse	0:00:05	A pop up window appears on the display	

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.33		Click on "MAKE CURRENT" in the pop up window			x					mouse, left display	0:00:05	The pop up window is removed from the display and an arrow appears to the left of the command	
1.3.34		Click on the arrow to the left of the command			x					mouse, left display	0:00:05	the arrow turns pink	
1.3.35		Verify once per second data				x				Left display	0:00:05	Once per second data	
1.3.36		Click on the arrow a second time			x					mouse, left display	0:00:05	The arrow turns blue	
1.3.37		Verify ENCDR value becomes correct (ENDCR2 = ON) on the TT&C page			x					Right display	0:00:15		
1.3.38		Read the next step in the pass plan. This calls for another 140 second wait.				x				left display	0:00:10		

			MODALITY										
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.39		Wait the prescribed amount of time				x				clock	0:02:20		Controller waits a bit more than 2 minutes.
1.3.40		Click on CONTINUE to go onto the next page of the pass plan			x					mouse, left display	0:00:05		
1.3.41		Read steps in the pass plan				x				left display	0:00:10		
1.3.42		Ask ARTS to go manual and configure for alternate downlink channel	x				x			phone	0:00:10	ARTS operator confirms change verbally	MC must remember the channel that came up originally and what channel to go to. This is trying to determine if the downlink transmitter is the problem
1.3.43		Click on arrow to go to next step			x					mouse. Left display	0:00:05		
1.3.44		Verify TMX =1 ON on the TT&C page				x				Right display	0:00:15		
1.3.45		Determine if ENCDR 1 is ON				x	x			Right display	0:00:15		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		(TT&C page)											
1.3.46		Click on the next step in the pass plan			x					mouse	0:00:05	A pop up window appears on the display	
1.3.47		Click on "MAKE CURRENT" in the pop up window			x					mouse, left display	0:00:05	The pop up window is removed from the display and an arrow appears to the left of the command	
1.3.48		Click on the arrow to the left of the command			x					mouse, left display	0:00:05	the arrow turns pink	
1.3.49		Verify once per second data				x				Left display	0:00:05	Once per second data	
1.3.50		Click on the arrow a second time			x					mouse, left display	0:00:05	The arrow turns blue	
1.3.51		Wait an additional 140 sec. (This wait occurs every time a component is switched.) Observe right				x				clock	0:02:20		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		display to determine if TLM starts											
1.3.52		Verify changes if TLM begins				x				Right display	0:00:15		
1.3.53		If still no TLM then go to next step (unencrypt data to see if encryptor is point of failure)					x						
1.3.54		Click on the next step in the pass plan			x					mouse	0:00:05	A pop up window appears on the display	
1.3.55		Click on "MAKE CURRENT" in the pop up window			x					mouse, left display	0:00:05	The pop up window is removed from the display and an arrow appears to the left of the command	
1.3.56		Click on the pop up window indicating that the step is			x					mouse, left display	0:00:05		Steps in the pass plan annotated with an "R" are restricted.

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
		restricted to indicate that you want to execute the step											
1.3.57		Click on the arrow to the left of the command			x					mouse, left display	0:00:05	the arrow turns pink	
1.3.58		Verify once per second data				x				. Left display	0:00:05	Once per second data	
1.3.59		Click on the arrow a second time			x					mouse, left display	0:00:05	The arrow turns blue	
1.3.60		Verify encryptors are in bypass on the TT&C page				x				Right screen	0:00:15	The values should read OFF. These values are in the third data column from the left in the TT&C Status page	
1.3.61		Read the next step in the pass plan				x				Left display	0:00:10		
1.3.62		Wait 140 sec. Observe TT&C to see if TLM begins to download				x				Right screen	0:02:20		

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE / DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.63		If TLM begins log problem.			x					scheduler computer			
1.3.64		Click on the end pass plan icon. If still no TLM contact SE			x					mouse	0:00:05	Clicking on the arrow will bring up th eoriginal pass plan.	The procedure did not fix the problem.
	End of of "No Telemetry " pass plan												

APPENDIX 9 - POST CONTACT SCORING SUMMARY

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.1	Score Contact												
1.1.1		Rate CONTACT as Success/Fail			x		x			mouse	0:00:05		
1.1.2		Rate COBRA as Success/Fail			x		x			mouse	0:00:05		
1.1.3		Rate AFSCN as Success/Fail			x		x			mouse	0:00:05		
1.2	Enter location of satellite at initial contact												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.1		Enter the difference in azimuth between predicted and actual			x					keyboard	0:00:05		data is used by Orbit Analysts. If the angular differences greater than about 1 degree indicates a potential problem in orbit prediction or in the satellite

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.2.2		Enter the difference in elevation between predicted and actual			x					keyboard	0:00:05		data is used by Orbit Analysts. If the angular differences greater than about 1 degree indicates a potential problem in orbit prediction or in the satellite
1.3	Elaborate												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.1		Enter notes on problems encountered with COBRA			x					keyboard	0:01:00		this step only done if there were problems. Typically, no problems are encountered and this step is skipped
1.3.2		Enter notes on problems encountered with AFSCN			x					keyboard	0:01:00		this step only done if there were problems. Typically, no problems are encountered and this step is skipped

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.3.3		Enter notes on problems encountered with the satellite			x					keyboard	0:01:00		this step only done if there were problems. Typically, no problems are encountered and this step is skipped

APPENDIX 10 - UPDATE SUPPORT SCHEDULE

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
	Boot System												
1.1.1		Type in user ID			x					keyboard	0:00:15		
1.1.2		Type in Password			x					keyboard	0:00:15		
1.1.3		Press the ENTER key in response to ready prompt			x					keyboard	0:00:02		
	Select Action												
1.2.1		Click on the "Deconfliction" menu option			x					mouse (keyboard alternate)	0:00:05	screen change	
1.2.2		Click on the "Run Macro" menu option			x					mouse (keyboard alternate)	0:00:05	screen change	
1.2.3		Click on the "Deconflict" menu option			x					mouse (keyboard alternate)	0:00:05	screen change	
	Select contact to be deconflicted												
1.3.1		Type in the time scheduled for the event to be deconflicted			x					keyboard	0:00:15		
	System												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
	action												
1.4.1		Wait for printer									0:00:30	hardcopy ejected from printer	
	Document												
1.5.1		Walk over to printer and get printout			x						0:00:10		
1.5.2		Put printout into the PAP binder			x						0:00:15		
	Examine Printout for Satellite A												
1.6.1		Compare requested time with time allocated				x	x				0:01:00		time from 1 to 5 minutes, depending on the number of conflicts and changes
1.6.2		Pencil in new times if they are in the acceptable window as defined in the PAP									0:00:15		time per change. This is repeated for each change
	Examine Printout for Satellite B												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.7.1		Compare requested time with time allocated				x	x				0:01:00		time from 1 to 5 minutes, depending on the number of conflicts and changes
1.7.2		Pencil in new times if they are in the acceptable window as defined in the PAP									0:00:15		time per change. This is repeated for each change
	Examine Printout for Satellite C												
1.8.1		Compare requested time with time allocated				x	x				0:01:00		time from 1 to 5 minutes, depending on the number of conflicts and changes
1.8.2		Pencil in new times if they are in the acceptable window as defined in the PAP									0:00:15		time per change. This is repeated for each change
	Examine Printout for Satellite D												

			MODALITY										
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.9.1		Compare requested time with time allocated				x	x				0:01:00		time from 1 to 5 minutes, depending on the number of conflicts and changes
1.9.2		Pencil in new times if they are in the acceptable window as defined in the PAP									0:00:15		time per change. This is repeated for each change
1.10	Call 22 SOPS												
1.10.1		Phone 22 SOPS, ask for person to do deconfliction	x	x							0:00:30		
1.10.2		Identify change of interest	x								0:00:30		repeated for each change
1.10.3		Accept or reject change proposed by 22 SOPS	x								0:00:10		repeated for each change
1.10.4		End phone call											
	Create hardcopy connfirming schedule changes												

					MODALITY								
TASK ID	TASK DESCRIPTION	SUBTASK DESCRIPTION	TALK	LISTEN	MANUAL	VISUAL	COGNITIVE/DECISION	INFO REQUIRED	INFO SOURCE	EQUIPMENT	TIME (hh:mm:ss)	FEEDBACK	NOTES
1.11.1		Click on "Create MSC"			x					mouse (keyboard backup)	0:00:05		time per item
1.11.2		Enter MSC			x					mouse (keyboard backup)	0:00:30		
1.11.3		Close Window			x					mouse (keyboard backup)	0:00:20		
1.11.4		Select "Send MSG"			x					mouse (keyboard backup)	0:00:05		
1.11.5		Select receipt to be printed			x					mouse (keyboard backup)	0:00:15		
1.11.6		Print receipt			x						0:00:15		
1.11.7		Pick up receipt from printer			x						0:00:10	hardcopy ejected from printer	

VOICE COMMAND LIST

TABLE OF CONTENTS

INTRODUCTION

This document contains all of the voice commands recognized by the satellite controller test bed. These commands allow the controller to request information about the satellite from the system, and to issue commands that effect the satellite.

CONTINUOUS VARIABLES

The voice commands allow the controller to access the values of continuous variables. There are three general ways in which these values are presented. First, the system can announce the current value of the variable. Second, the system can display the value of the variable graphically. Thirdly, the system can both announce the value and display the value graphically. As the voice recognition system does not perform natural language recognition, a means of making the intent of the controller clear to the system is needed. Therefore, a standard phraseology convention was required. Here, commands that begin with the phrase "Tell me …" will cause the system to announce the value of the variable. Commands that begin with the words "Show me …" cause the system to display the value on the screen. Finally, commands that begin with the words "What is…" or simply contain only the name of the variable or a description of the measure (e.g., "internal temperature of the digital telemetry unit") will cause the system to both announce and display the value of that variables.

We have found that users quickly adapt to this phraseology convention and do not find that it is so unnatural that it is burdensome or interferes with their interaction with the test bed.

For each variable, there are a number of phrases that begin with "Tell me …", "Show me …", and "What is …". These phrases are synonymous. That is, they allow the controller some latitude in the terminology used to cause an effect. This allows the controller to interact with the system as if it were performing some level of natural language recognition.

Below is an example of a voice command.

vs_C_PLUS_150V - "Show me see plus one fifty vee"

The first three characters of each line, which are "vs_" in this example, indicate the type of response expected when the command is given. "vs_" indicates that the system will respond by displaying a graphic with the value, "vt_" indicates that the system will respond by announcing the value, and "vw_" indicates that the system will both display and announce the value. This is followed by the name of the variable. Finally, the text in quotes is the phrase that is recognized.

DISCRETE VARIABLES

Discrete variables generally represent controls aboard the satellite that the controller may manipulate. These discrete variables include switches (ON or OFF, encrypt or don't encrypt) and

power settings (2.5 watt and 20 watt), as examples. In the test bed the controller can activate one of these controls by making a voice command, as well as by executing a pass plan or by manually setting the variable in the graphic display of the sub-system.

As in the case of continuous variables, the controller has the option of saying any one of the synonymous phrases to obtain the desired consequence. That is, for example, the controller could say either "set ee dee tea you bee bee to on" or "set digital telemetry unit bee to on" or "turn on digital telemetry unit bee" and have variable EDTUBB change from OFF to ON. This latitude eliminates the need for the controller to use a rigid syntax.

Note that a controller may use descriptive terms (e.g., "digital telemetry unit B") to interact with the satellite, rather than requiring the controller to know, recall, and use a six letter alphanumeric label that may have little or no intrinsic meaning. Allowing the use of descriptive terminology rather than requiring use of the variable's label is intended to allow persons to perform a support even when their ability to recall and use the names of the variables has decayed through lack of use. (It is easy to predict that this will be the case in situations involving largely unattended operations.)

Below is an example of a command that effects a discrete variable.

C2ASPB_1 - "change the antenna switch position monitor from eh high to bee high"

The first part of the example indicates the variable that is being changed, and the state it is being changed to. In this example, a value of "1" indicates a state of "B-high". The portion of the command in quotes is the phrase spoken by the controller and recognized by the voice recognition system.

SUB-SYSTEM DIAGRAMS

There are a number of sub-system diagrams available for display. The controller may use verbal commands to display these diagrams as well as to remove them from the workstation.

PASS PLANS

The controller may use verbal commands to bring up or remove any of the available pass plans from the displays on the workstation.

CONTINUOUS COMMANDS

LINK 2 COMMUNICATIONS SUBSYSTEM

C+150V

vw_C_PLUS_150V - "see plus one fifty vee"
vw_C_PLUS_150V - "see plus one hundred fifty vee"
vw_C_PLUS_150V - "see plus one five owe vee"
vw_C_PLUS_150V - "see plus one five zero vee"
vw_C_PLUS_150V - "voltage of the digital telemetry unit"
vw_C_PLUS_150V - "digital telemetry unit's voltage"
vs_C_PLUS_150V - "Show me see plus one fifty vee"
vt_C_PLUS_150V - "Tell me see plus one fifty vee"
vw_C_PLUS_150V - "What is see plus one fifty vee"
vs_C_PLUS_150V - "show me see plus one hundred fifty vee"
vt_C_PLUS_150V - "tell me see plus one hundred fifty vee"
vw_C_PLUS_150V - "what is see plus one hundred fifty vee"
vs_C_PLUS_150V - "show me see plus one five owe vee"
vt_C_PLUS_150V - "tell me see plus one five owe vee"
vw_C_PLUS_150V - "what is see plus one five owe vee"
vs_C_PLUS_150V - "show me see plus one five zero vee"
vt_C_PLUS_150V - "tell me see plus one five zero vee"
vw_C_PLUS_150V - "what is see plus one five zero vee"
vs_C_PLUS_150V - "Show me the value of see plus one fifty vee"
vt_C_PLUS_150V - "Tell me the value of see plus one fifty vee"
vw_C_PLUS_150V - "What is the value of see plus one fifty vee"
vs_C_PLUS_150V - "show me the value of see plus one hundred fifty vee"
vt_C_PLUS_150V - "tell me the value of see plus one hundred fifty vee"
vw_C_PLUS_150V - "what is the value of see plus one hundred fifty vee"
vs_C_PLUS_150V - "show me the value of see plus one five owe vee"
vt_C_PLUS_150V - "tell me the value of see plus one five owe vee"

vw_C_PLUS_150V - "what is the value of see plus one five owe vee"
vs_C_PLUS_150V - "show me the value of see plus one five zero vee"
vt_C_PLUS_150V - "tell me the value of see plus one five zero vee"
vw_C_PLUS_150V - "what is the value of see plus one five zero vee"
vs_C_PLUS_150V - "Show me the voltage of the digital telemetry unit"
vt_C_PLUS_150V - "Tell me the voltage of the digital telemetry unit"
vw_C_PLUS_150V - "What is the voltage of the digital telemetry unit"
vs_C_PLUS_150V - "Show me the digital telemetry unit's voltage"
vt_C_PLUS_150V - "Tell me the digital telemetry unit's voltage"
vw_C_PLUS_150V - "What is the digital telemetry unit's voltage"

<u>C2A+1V</u>
vw_C2A_PLUS_1V - "see two eh plus one vee"
vw_C2A_PLUS_1V - "link two eh positive fifteen volt output voltage"
vw_C2A_PLUS_1V - "voltage of the link two eh positive fifteen volt output"
vs_C2A_PLUS_1V - "Show me see two eh plus one vee"
vt_C2A_PLUS_1V - "Tell me see two eh plus one vee"
vw_C2A_PLUS_1V - "What is see two eh plus one vee"
vs_C2A_PLUS_1V - "Show me the link two eh positive fifteen volt output voltage"
vs_C2A_PLUS_1V - "Show me the value of see two eh plus one vee"
vt_C2A_PLUS_1V - "Tell me the value of see two eh plus one vee"
vw_C2A_PLUS_1V - "What is the value of see two eh plus one vee"
vs_C2A_PLUS_1V - "Show me the link two eh positive fifteen volt output voltage"
vt_C2A_PLUS_1V - "Tell me the link two eh positive fifteen volt output voltage"
vw_C2A_PLUS_1V - "What is the link two eh positive fifteen volt output voltage"
vs_C2A_PLUS_1V - "Show me the voltage of the link two eh positive fifteen volt output"
vt_C2A_PLUS_1V - "Tell me the voltage of the link two eh positive fifteen volt output"
vw_C2A_PLUS_1V - "What is the voltage of the link two eh positive fifteen volt output"

<u>C2A-1V</u>
vw_C2A_MINUS_1V - "see two eh minus one vee"
vw_C2A_MINUS_1V - "link two eh negative fifteen volt output voltage"
vw_C2A_MINUS_1V - "voltage of the link two eh negative fifteen volt output"
vs_C2A_MINUS_1V - "Show me see two eh minus one vee"

vt_C2A_MINUS_1V - "Tell me see two eh minus one vee"
vw_C2A_MINUS_1V - "What is see two eh minus one vee"
vs_C2A_MINUS_1V - "Show me the value of see two eh minus one vee"
vt_C2A_MINUS_1V - "Tell me the value of see two eh minus one vee"
vw_C2A_MINUS_1V - "What is the value of see two eh minus one vee"
vs_C2A_MINUS_1V - "Show me the link two eh negative volt output voltage"
vt_C2A_MINUS_1V - "Tell me the link two eh negative fifteen volt output voltage"
vw_C2A_MINUS_1V - "What is the link two eh negative fifteen volt output voltage"
vs_C2A_MINUS_1V - "Show me the voltage of the link two eh negative fifteen volt output"
vt_C2A_MINUS_1V - "Tell me the voltage of the link two eh negative fifteen volt output"
vw_C2A_MINUS_1V - "What is the voltage of the link two eh negative fifteen volt output"

<u>C2A28V</u>
vw_C2A28V - "see two eh twenty eight vee"
vw_C2A28V - "see two eh two eight vee"
vw_C2A28V - "link two eh positive twenty eight volt output voltage"
vw_C2A28V - "voltage of the link two eh positive twenty eight volt output"
vs_C2A28V - "Show me see two eh twenty eight vee"
vt_C2A28V - "Tell me see two eh twenty eight vee"
vw_C2A28V - "What is see two eh twenty eight vee"
vs_C2A28V - "Show me see two eh two eight vee"
vt_C2A28V - "Tell me see two eh two eight vee"
vw_C2A28V - "What is see two eh two eight vee"
vs_C2A28V - "Show me the value of see two eh twenty eight vee"
vt_C2A28V - "Tell me the value of see two eh twenty eight vee"
vw_C2A28V - "What is the value of see two eh twenty eight vee"
vs_C2A28V - "Show me the value of see two eh two eight vee"
vt_C2A28V - "Tell me the value of see two eh two eight vee"
vw_C2A28V - "What is the value of see two eh two eight vee"
vs_C2A28V - "Show me the link two eh positive twenty eight volt output voltage"
vt_C2A28V - "Tell me the link two eh positive twenty eight volt output voltage"
vw_C2A28V - "What is the link two eh positive twenty eight volt output voltage"
vs_C2A28V - "Show me the voltage of the link two eh positive twenty eight volt output"
vt_C2A28V - "Tell me the voltage of the link two eh positive twenty eight volt output"

vw_C2A28V - "What is the voltage of the link two eh positive twenty eight volt output"

C2B+1V

vw_C2B_PLUS_1V - "see two bee plus one vee"
vw_C2B_PLUS_1V - "link two bee positive fifteen volt output voltage"
vw_C2B_PLUS_1V - "voltage of the link two bee positive fifteen volt output"
vs_C2B_PLUS_1V - "Show me see two bee plus one vee"
vt_C2B_PLUS_1V - "Tell me see two bee plus one vee"
vw_C2B_PLUS_1V - "What is see two bee plus one vee"
vs_C2B_PLUS_1V - "Show me the value of see two bee plus one vee"
vt_C2B_PLUS_1V - "Tell me the value of see two bee plus one vee"
vw_C2B_PLUS_1V - "What is the value of see two bee plus one vee"
vs_C2B_PLUS_1V - "Show me the link two bee positive fifteen volt output voltage"
vt_C2B_PLUS_1V - "Tell me the link two bee positive fifteen volt output voltage"
vw_C2B_PLUS_1V - "What is the link two bee positive fifteen volt output voltage"
vs_C2B_PLUS_1V - "Show me the voltage of the link two bee positive fifteen volt output"
vt_C2B_PLUS_1V - "Tell me the voltage of the link two bee positive fifteen volt output"
vw_C2B_PLUS_1V - "What is the voltage of the link two bee positive fifteen volt output"

C2B-1V

vw_C2B_MINUS_1V - "see two bee minus one vee"
vw_C2B_MINUS_1V - "link two bee negative fifteen volt output voltage"
vw_C2B_MINUS_1V - "voltage of the link two bee negative fifteen volt output"
vs_C2B_MINUS_1V - "Show me see two bee minus one vee"
vt_C2B_MINUS_1V - "Tell me see two bee minus one vee"
vw_C2B_MINUS_1V - "What is see two bee minus one vee"
vs_C2B_MINUS_1V - "Show me the value of see two bee minus one vee"
vt_C2B_MINUS_1V - "Tell me the value of see two bee minus one vee"
vw_C2B_MINUS_1V - "What is the value of see two bee minus one vee"
vs_C2B_MINUS_1V - "Show me the link two bee negative fifteen volt output voltage"
vt_C2B_MINUS_1V - "Tell me the link two bee negative fifteen volt output voltage"
vw_C2B_MINUS_1V - "What is the link two bee negative fifteen volt output voltage"
vs_C2B_MINUS_1V - "Show me the voltage of the link two bee negative fifteen volt output"
vt_C2B_MINUS_1V - "Tell me the voltage of the link two bee negative fifteen volt output"

vw_C2B_MINUS_1V - "What is the voltage of the link two bee negative fifteen volt output"

C2B28V
vw_C2B28V - "see two bee twenty eight vee"
vw_C2B28V - "see two bee two eight vee"
vw_C2B28V - "link two bee positive twenty eight volt output voltage"
vw_C2B28V - "voltage of the link two bee positive twenty eight volt output"
vs_C2B28V - "Show me see two bee twenty eight vee"
vt_C2B28V - "Tell me see two bee twenty eight vee"
vw_C2B28V - "What is see two bee twenty eight vee"
vs_C2B28V - "Show me the value of see two bee twenty eight vee"
vt_C2B28V - "Tell me the value of see two bee twenty eight vee"
vw_C2B28V - "What is the value of see two bee twenty eight vee"
vs_C2B28V - "Show me see two bee two eight vee"
vt_C2B28V - "Tell me see two bee two eight vee"
vw_C2B28V - "What is see two bee two eight vee"
vs_C2B28V - "Show me the value of see two bee two eight vee"
vt_C2B28V - "Tell me the value of see two bee two eight vee"
vw_C2B28V - "What is the value of see two bee two eight vee"
vs_C2B28V - "Show me the link two bee positive twenty eight volt output voltage"
vt_C2B28V - "Tell me the link two bee positive twenty eight volt output voltage"
vw_C2B28V - "What is the link two bee positive twenty eight volt output voltage"
vs_C2B28V - "Show me the voltage of the link two bee positive twenty eight volt output"
vt_C2B28V - "Tell me the voltage of the link two bee positive twenty eight volt output"
vw_C2B28V - "What is the voltage of the link two bee positive twenty eight volt output"

CC+50V
vw_CC_PLUS_50V - "see see plus fifty vee"
vw_CC_PLUS_50V - "see see plus five zero vee"
vw_CC_PLUS_50V - "see see plus five oh vee"
vw_CC_PLUS_50V - "digital telemetry unit's positive five volt output voltage"
vw_CC_PLUS_50V - "voltage of the digital telemetry unit's positive five volt output"
vs_CC_PLUS_50V - "Show me see see plus fifty vee"
vt_CC_PLUS_50V - "Tell me see see plus fifty vee"

vw_CC_PLUS_50V - "What is see see plus fifty vee"
vs_CC_PLUS_50V - "Show me see see plus five zero vee"
vt_CC_PLUS_50V - "Tell me see see plus five zero vee"
vw_CC_PLUS_50V - "What is see see plus five zero vee"
vs_CC_PLUS_50V - "Show me see see plus five oh vee"
vt_CC_PLUS_50V - "Tell me see see plus five oh vee"
vw_CC_PLUS_50V - "What is see see plus five oh vee"
vs_CC_PLUS_50V - "Show me the value of see see plus fifty vee"
vt_CC_PLUS_50V - "Tell me the value of see see plus fifty vee"
vw_CC_PLUS_50V - "What is the value of see see plus fifty vee"
vs_CC_PLUS_50V - "Show me the value of see see plus five zero vee"
vt_CC_PLUS_50V - "Tell me the value of see see plus five zero vee"
vw_CC_PLUS_50V - "What is the value of see see plus five zero vee"
vs_CC_PLUS_50V - "Show me the value of see see plus five oh vee"
vt_CC_PLUS_50V - "Tell me the value of see see plus five oh vee"
vw_CC_PLUS_50V - "What is the value of see see plus five oh vee"
vs_CC_PLUS_50V - "Show me the digital telemetry unit's positive five volt output voltage"
vt_CC_PLUS_50V - "Tell me the digital telemetry unit's positive five volt output voltage"
vw_CC_PLUS_50V - "What is the digital telemetry unit's positive five volt output voltage"
vs_CC_PLUS_50V - "Show me the voltage of the digital telemetry unit's positive five volt output"
vt_CC_PLUS_50V - "Tell me the voltage of the digital telemetry unit's positive five volt output"
vw_CC_PLUS_50V - "What is the voltage of the digital telemetry unit's positive five volt output"

CC+60V
vw_CC_PLUS_60V - "see see plus sixty vee"
vw_CC_PLUS_60V - "see see plus six zero vee"
vw_CC_PLUS_60V - "see see plus six oh vee"
vw_CC_PLUS_60V - "digital telemetry unit's positive six volt output voltage"
vw_CC_PLUS_60V - "voltage of the digital telemetry unit's positive six volt output"
vs_CC_PLUS_60V - "Show me see see plus sixty vee"
vt_CC_PLUS_60V - "Tell me see see plus sixty vee"
vw_CC_PLUS_60V - "What is see see plus sixty vee"
vs_CC_PLUS_60V - "Show me see see plus six zero vee"
vt_CC_PLUS_60V - "Tell me see see plus six zero vee"

vw_CC_PLUS_60V - "What is see see plus six zero vee"
vs_CC_PLUS_60V - "Show me see see plus six oh vee"
vt_CC_PLUS_60V - "Tell me see see plus six oh vee"
vw_CC_PLUS_60V - "What is see see plus six oh vee"
vs_CC_PLUS_60V - "Show me the value of see see plus sixty vee"
vt_CC_PLUS_60V - "Tell me the value of see see plus sixty vee"
vw_CC_PLUS_60V - "What is the value of see see plus sixty vee"
vs_CC_PLUS_60V - "Show me the value of see see plus six zero vee"
vt_CC_PLUS_60V - "Tell me the value of see see plus six zero vee"
vw_CC_PLUS_60V - "What is the value of see see plus six zero vee"
vs_CC_PLUS_60V - "Show me the value of see see plus six oh vee"
vt_CC_PLUS_60V - "Tell me the value of see see plus six oh vee"
vw_CC_PLUS_60V - "What is the value of see see plus six oh vee"
vs_CC_PLUS_60V - "Show me the digital telemetry unit's positive six volt output voltage"
vt_CC_PLUS_60V - "Tell me the digital telemetry unit's positive six volt output voltage"
vw_CC_PLUS_60V - "What is the digital telemetry unit's positive six volt output voltage"
vs_CC_PLUS_60V - "Show me the voltage of the digital telemetry unit's positive six volt output"
vt_CC_PLUS_60V - "Tell me the voltage of the digital telemetry unit's positive six volt output"
vw_CC_PLUS_60V - "What is the voltage of the digital telemetry unit's positive six volt output"

CC-23V

vw_CC_MINUS_23V - "sea sea minus twenty three vee"
vw_CC_MINUS_23V - "sea sea minus two three vee"
vw_CC_MINUS_23V - "digital telemetry unit's negative twenty three point five volt output voltage"
vw_CC_MINUS_23V - "digital telemetry unit's negative twenty three and a half volt output voltage"
vs_CC_MINUS_23V - "Show me sea sea minus twenty three vee"
vt_CC_MINUS_23V - "Tell me sea sea minus twenty three vee"
vw_CC_MINUS_23V - "What is sea sea minus twenty three vee"
vs_CC_MINUS_23V - "Show me sea sea minus two three vee"
vt_CC_MINUS_23V - "Tell me sea sea minus two three vee"
vw_CC_MINUS_23V - "What is sea sea minus two three vee"
vs_CC_MINUS_23V - "Show me the value of sea sea minus twenty three vee"
vt_CC_MINUS_23V - "Tell me the value of sea sea minus twenty three vee"
vw_CC_MINUS_23V - "What is the value of sea sea minus twenty three vee"

vs_CC_MINUS_23V - "Show me the value of sea sea minus two three vee"
vt_CC_MINUS_23V - "Tell me the value of sea sea minus two three vee"
vw_CC_MINUS_23V - "What is the value of sea sea minus two three vee"
vs_CC_MINUS_23V - "Show me the digital telemetry unit's negative twenty three point five volt output voltage"
vt_CC_MINUS_23V - "Tell me the digital telemetry unit's negative twenty three point five volt output voltage"
vw_CC_MINUS_23V - "What is the digital telemetry unit's negative twenty three point five volt output voltage"
vs_CC_MINUS_23V - "Show me the digital telemetry unit's negative twenty three and a half volt output voltage"
vt_CC_MINUS_23V - "What is the digital telemetry unit's negative twenty three and a half volt output voltage"
vw_CC_MINUS_23V - "Show me the digital telemetry unit's negative twenty three and a half volt output voltage"

<u>CC-60V</u>
vw_CC_MINUS_60V - "sea sea minus sixty vee"
vw_CC_MINUS_60V - "sea sea minus six zero vee"
vw_CC_MINUS_60V - "sea sea minus six oh vee"
vw_CC_MINUS_60V - "digital telemetry unit's negative six volt output voltage"
vs_CC_MINUS_60V - "Show me sea sea minus sixty vee"
vt_CC_MINUS_60V - "Tell me sea sea minus sixty vee"
vw_CC_MINUS_60V - "What is sea sea minus sixty vee"
vs_CC_MINUS_60V - "Show me sea sea minus six zero vee"
vt_CC_MINUS_60V - "Tell me sea sea minus six zero vee"
vw_CC_MINUS_60V - "What is sea sea minus six zero vee"
vs_CC_MINUS_60V - "Show me sea sea minus six oh vee"
vt_CC_MINUS_60V - "Tell me sea sea minus six oh vee"
vw_CC_MINUS_60V - "What is sea sea minus six oh vee"
vs_CC_MINUS_60V - "Show me the value of sea sea minus sixty vee"
vt_CC_MINUS_60V - "Tell me the value of sea sea minus sixty vee"
vw_CC_MINUS_60V - "What is the value of sea sea minus sixty vee"
vs_CC_MINUS_60V - "Show me the value of sea sea minus six zero vee"
vt_CC_MINUS_60V - "Tell me the value of sea sea minus six zero vee"
vw_CC_MINUS_60V - "What is the value of sea sea minus six zero vee"
vs_CC_MINUS_60V - "Show me sea the value of sea minus six oh vee"
vt_CC_MINUS_60V - "Tell me the value of sea sea minus six oh vee"
vw_CC_MINUS_60V - "What is the value of sea sea minus six oh vee"
vs_CC_MINUS_60V - "Show me the digital telemetry unit's negative six volt output voltage"

vt_CC_MINUS_60V - "Tell me the digital telemetry unit's negative six volt output voltage"
vw_CC_MINUS_60V - "What is the digital telemetry unit's negative six volt output voltage"

CCINIT
vw_CCINIT - "sea sea INIT"
vw_CCINIT - "sea sea eye en eye tea"
vw_CCINIT - "internal temperature of the digital telemetry unit"
vs_CCINIT - "Show me sea sea INIT"
vt_CCINIT - "Tell me sea sea INIT"
vw_CCINIT - "What is sea sea INIT"
vs_CCINIT - "Show me sea sea eye en eye tea"
vt_CCINIT - "Tell me sea sea eye en eye tea"
vw_CCINIT - "What is sea sea eye en eye tea"
vs_CCINIT - "Show me the value of sea sea INIT"
vt_CCINIT - "Tell me the value of sea sea INIT"
vw_CCINIT - "What is the value of sea sea INIT"
vs_CCINIT - "Show me the value of sea sea eye en eye tea"
vt_CCINIT - "Tell me the value of sea sea eye en eye tea"
vw_CCINIT - "What is the value of sea sea eye en eye tea"
vs_CCINIT - "Show me the internal temperature of the digital telemetry unit"
vt_CCINIT - "Tell me the internal temperature of the digital telemetry unit"
vw_CCINIT - "What is the internal temperature of the digital telemetry unit"

CCS1CV
vw_CCS1CV - "sea sea ess one sea vee"
vw_CCS1CV - "calibration voltage one"
vs_CCS1CV - "Show me sea sea ess one sea vee"
vt_CCS1CV - "Tell me sea sea ess one sea vee"
vw_CCS1CV - "What is sea sea ess one sea vee"
vs_CCS1CV - "Show me the value of sea sea ess one sea vee"
vt_CCS1CV - "Tell me the value of sea sea ess one sea vee"
vw_CCS1CV - "What is the value of sea sea ess one sea vee"
vs_CCS1CV - "Show me calibration voltage one"
vt_CCS1CV - "Tell me calibration voltage one"

vw_CCS1CV - "What is calibration voltage one"

CCS2CV

vw_CCS2CV - "sea sea ess two sea vee"
vw_CCS2CV - "calibration voltage two"
vs_CCS2CV - "Show me sea sea ess two sea vee"
vt_CCS2CV - "Tell me sea sea ess two sea vee"
vw_CCS2CV - "What is sea sea ess two sea vee"
vs_CCS2CV - "Show me the value of sea sea ess two sea vee"
vt_CCS2CV - "Tell me the value of sea sea ess two sea vee"
vw_CCS2CV - "What is the value of sea sea ess two sea vee"
vs_CCS2CV - "Show me calibration voltage two"
vt_CCS2CV - "Tell me calibration voltage two"
vw_CCS2CV - "What is calibration voltage two"

CCS3CV

vw_CCS3CV - "sea sea ess three sea vee"
vw_CCS3CV - "calibration voltage three"
vs_CCS3CV - "Show me sea sea ess three sea vee"
vt_CCS3CV - "Tell me sea sea ess three sea vee"
vw_CCS3CV - "What is sea sea ess three sea vee"
vs_CCS3CV - "Show me the value of sea sea ess three sea vee"
vt_CCS3CV - "Tell me the value of sea sea ess three sea vee"
vw_CCS3CV - "What is the value of sea sea ess three sea vee"
vs_CCS3CV - "Show me calibration voltage three"
vt_CCS3CV - "Tell me calibration voltage three"
vw_CCS3CV - "What is calibration voltage three"

MISCELLANEOUS COMMANDS AND VARIABLES

ShowSumTransmtrA - "Show me a summary of Transmitter A"
TellSumTransmtrA - "Tell me a summary of Transmitter A"
ShowSumTransmtrA - "Show me a summary of Transmitter eh

TellSumTransmtrA - "Tell me a summary of Transmitter eh"
ShowSumTransmtrB - "Show me a summary of Transmitter B"
TellSumTransmtrB - "Tell me a summary of Transmitter B"
ShowSumTransmtrB - "Show me a summary of Transmitter bee"
TellSumTransmtrB - "Tell me a summary of Transmitter bee"

LINK 1 COMMUNICATIONS SUBSYSTEM

CCT1AV

vw_CCT1AV - "sea sea tea one eh vee"
vw_CCT1AV - "voltage of transmitter eh"
vw_CCT1AV - "output voltage of transmitter eh"
vw_CCT1AV - "transmitter eh's output voltage"
vs_CCT1AV - "Show me sea sea tea one eh vee"
vt_CCT1AV - "Tell me sea sea tea one eh vee"
vw_CCT1AV - "What is sea sea tea one eh vee"
vs_CCT1AV - "Show me the value of sea sea tea one eh vee"
vt_CCT1AV - "Tell me the value of sea sea tea one eh vee"
vw_CCT1AV - "What is the value of sea sea tea one eh vee"
vs_CCT1AV - "Show me the voltage of transmitter eh"
vt_CCT1AV - "Tell me the voltage of transmitter eh"
vw_CCT1AV - "What is the voltage of transmitter eh"
vs_CCT1AV - "Show me the output voltage of transmitter eh"
vt_CCT1AV - "Tell me the output voltage of transmitter eh"
vw_CCT1AV - "What is the output voltage of transmitter eh"
vs_CCT1AV - "Show me transmitter eh's output voltage"
vt_CCT1AV - "Tell me transmitter eh's output voltage"
vw_CCT1AV - "What is transmitter eh's output voltage"

CCT1AW

vw_CCT1AW - "sea sea tea one eh double you"
vw_CCT1AW - "transmitter eh's are eff power"
vw_CCT1AW - "are eff power of transmitter eh"

vs_CCT1AW - "Show me sea sea tea one eh double you"
vt_CCT1AW - "Tell me sea sea tea one eh double you"
vw_CCT1AW - "What is sea sea tea one eh double you"
vs_CCT1AW - "Show me the value of sea sea tea one eh double you"
vt_CCT1AW - "Tell me the value of sea sea tea one eh double you"
vw_CCT1AW - "What is the value of sea sea tea one eh double you"
vs_CCT1AW - "Show me transmitter eh's are eff power"
vt_CCT1AW - "Tell me transmitter eh's are eff power"
vw_CCT1AW - "What is transmitter eh's are eff power"
vs_CCT1AW - "Show me the are eff power of transmitter eh"
vt_CCT1AW - "Tell me the are eff power of transmitter eh
vw_CCT1AW - "What is the are eff power of transmitter eh"

<u>CCT1BV</u>
vw_CCT1BV - "sea sea tea one bee vee"
vw_CCT1BV - "voltage of transmitter bee"
vw_CCT1BV - "output voltage of transmitter bee"
vw_CCT1BV - "transmitter bee's output voltage"
vs_CCT1BV - "Show me sea sea tea one bee vee"
vt_CCT1BV - "Tell me sea sea tea one bee vee"
vw_CCT1BV - "What is sea sea tea one bee vee"
vs_CCT1BV - "Show me the value of sea sea tea one bee vee"
vt_CCT1BV - "Tell me the value of sea sea tea one bee vee"
vw_CCT1BV - "What is the value of sea sea tea one bee vee"
vs_CCT1BV - "Show me the voltage of transmitter bee"
vt_CCT1BV - "Tell me the voltage of transmitter bee"
vw_CCT1BV - "What is the voltage of transmitter bee"
vs_CCT1BV - "Show me the output voltage of transmitter bee"
vt_CCT1BV - "Tell me the output voltage of transmitter bee"
vw_CCT1BV - "What is the output voltage of transmitter bee"
vs_CCT1BV - "Show me transmitter bee's output voltage"
vt_CCT1BV - "Tell me transmitter bee's output voltage"
vw_CCT1BV - "What is transmitter bee's output voltage"

CCT1BW

vw_CCT1BW - "sea sea tea one bee double you"
vw_CCT1BW - "transmitter bee's are eff power"
vw_CCT1BW - "are eff power of transmitter bee"
vs_CCT1BW - "Show me sea sea tea one bee double you"
vt_CCT1BW - "Tell me sea sea tea one bee double you"
vw_CCT1BW - "What is sea sea tea one bee double you"
vs_CCT1BW - "Show me the value of sea sea tea one bee double you"
vt_CCT1BW - "Tell me the value of sea sea tea one bee double you"
vw_CCT1BW - "What is the value of sea sea tea one bee double you"
vs_CCT1BW - "Show me transmitter bee's are eff power"
vt_CCT1BW - "Tell me transmitter bee's are eff power"
vw_CCT1BW - "What is transmitter bee's are eff power"
vs_CCT1BW - "Show me the are eff power of transmitter bee"
vt_CCT1BW - "Tell me the are eff power of transmitter bee"
vw_CCT1BW - "What is the are eff power of transmitter bee"

CECA5V

vw_CECA5V - "sea ee sea eh five vee"
vw_CECA5V - "voltage of dual error coder eh"
vw_CECA5V - "dual error coder eh's voltage"
vs_CECA5V - "Show me sea ee sea eh five vee"
vt_CECA5V - "Tell me sea ee sea eh five vee"
vw_CECA5V - "What is sea ee sea eh five vee"
vs_CECA5V - "Show me the value of sea ee sea eh five vee"
vt_CECA5V - "Tell me the value of sea ee sea eh five vee"
vw_CECA5V - "What is the value of sea ee sea eh five vee"
vs_CECA5V - "Show me the voltage of dual error coder eh"
vt_CECA5V - "Tell me the voltage of dual error coder eh"
vw_CECA5V - "What is the voltage of dual error coder eh"
vs_CECA5V - "Show me dual error coder eh's voltage"
vt_CECA5V - "Tell me dual error coder eh's voltage"
vw_CECA5V - "What is dual error coder eh's voltage"

CECB5V

vw_CECB5V - "sea ee sea bee five vee"
vw_CECB5V - "voltage of dual error coder bee"
vw_CECB5V - "dual error coder bee's voltage"
vs_CECB5V - "Show me sea ee sea bee five vee"
vt_CECB5V - "Tell me sea ee sea bee five vee"
vw_CECB5V - "What is sea ee sea bee five vee"
vs_CECB5V - "Show me the value of sea ee sea bee five vee"
vt_CECB5V - "Tell me the value of sea ee sea bee five vee"
vw_CECB5V - "What is the value of sea ee sea bee five vee"
vs_CECB5V - "Show me the voltage of dual error coder bee"
vt_CECB5V - "Tell me the voltage of dual error coder bee"
vw_CECB5V - "What is the voltage of dual error coder bee"
vs_CECB5V - "Show me dual error coder bee's voltage"
vt_CECB5V - "Tell me dual error coder bee's voltage"
vw_CECB5V - "What is dual error coder bee's voltage"

CPA1AT

vw_CPA1AT - "sea pea eh one eh tea"
vw_CPA1AT - "eh temperature of the solid state power amplifier"
vw_CPA1AT - "solid state power amplifier's eh temperature"
vs_CPA1AT - "Show me sea pea eh one eh tea"
vt_CPA1AT - "Tell me sea pea eh one eh tea"
vw_CPA1AT - "What is sea pea eh one eh tea"
vs_CPA1AT - "Show me the value of sea pea eh one eh tea"
vt_CPA1AT - "Tell me the value of sea pea eh one eh tea"
vw_CPA1AT - "What is the value of sea pea eh one eh tea"
vs_CPA1AT - "Show me the eh temperature of the solid state power amplifier"
vt_CPA1AT - "Tell me the eh temperature of the solid state power amplifier"
vw_CPA1AT - "What is the eh temperature of the solid state power amplifier"
vs_CPA1AT - "Show me the solid state power amplifier's eh temperature"
vt_CPA1AT - "Tell me the solid state power amplifier's eh temperature"
vw_CPA1AT - "What is the solid state power amplifier's eh temperature"

CPA1AV

vw_CPA1AV - "sea pea eh one eh vee"
vw_CPA1AV - "voltage of solid state power amplifier eh"
vw_CPA1AV - "solid state power amplifier eh's voltage"
vs_CPA1AV - "Show me sea pea eh one eh vee"
vt_CPA1AV - "Tell me sea pea eh one eh vee"
vw_CPA1AV - "What is sea pea eh one eh vee"
vs_CPA1AV - "Show me the value of sea pea eh one eh vee"
vt_CPA1AV - "Tell me the value of sea pea eh one eh vee"
vw_CPA1AV - "What is the value of sea pea eh one eh vee"
vs_CPA1AV - "Show me the voltage of solid state power amplifier eh"
vt_CPA1AV - "Tell me the voltage of solid state power amplifier eh"
vw_CPA1AV - "What is the voltage of solid state power amplifier eh"
vs_CPA1AV - "Show me solid state power amplifier eh's voltage"
vt_CPA1AV - "Tell me solid state power amplifier eh's voltage"
vw_CPA1AV - "What is solid state power amplifier eh's voltage"

CPA1AW

vw_CPA1AW - "sea pea eh one eh double you"
vw_CPA1AW - "are eff power of solid state amplifier eh"
vw_CPA1AW - "solid state amplifier eh's are eff power"
vs_CPA1AW - "Show me sea pea eh one eh double you"
vt_CPA1AW - "Tell me sea pea eh one eh double you"
vw_CPA1AW - "What is sea pea eh one eh double you"
vs_CPA1AW - "Show me the value of sea pea eh one eh double you"
vt_CPA1AW - "Tell me the value of sea pea eh one eh double you"
vw_CPA1AW - "What is the value of sea pea eh one eh double you"
vs_CPA1AW - "Show me the are eff power of solid state amplifier eh"
vt_CPA1AW - "Tell me the are eff power of solid state amplifier eh"
vw_CPA1AW - "What is the are eff power of solid state amplifier eh"
vs_CPA1AW - "Show me solid state amplifier eh's are eff power"
vt_CPA1AW - "Tell me solid state amplifier eh's are eff power"
vw_CPA1AW - "What is solid state amplifier eh's are eff power"

CPA1BT

vw_CPA1BT - "sea pea eh one bee tea"
vw_CPA1BT - "temperature of solid state power amplifier bee"
vw_CPA1BT - "solid state power amplifier bee's temperature"
vs_CPA1BT - "Show me sea pea eh one bee tea"
vt_CPA1BT - "Tell me sea pea eh one bee tea"
vw_CPA1BT - "What is sea pea eh one bee tea"
vs_CPA1BT - "Show me the value of sea pea eh one bee tea"
vt_CPA1BT - "Tell me the value of sea pea eh one bee tea"
vw_CPA1BT - "What is the value of sea pea eh one bee tea"
vs_CPA1BT - "Show me the temperature of solid state power amplifier bee"
vt_CPA1BT - "Tell me the temperature of solid state power amplifier bee"
vw_CPA1BT - "What is the temperature of solid state power amplifier bee"
vs_CPA1BT - "Show me solid state power amplifier bee's temperature"
vt_CPA1BT - "Tell me solid state power amplifier bee's temperature"
vw_CPA1BT - "What is solid state power amplifier bee's temperature"

CPA1BV

vw_CPA1BV - "sea pea eh one bee vee"
vw_CPA1BV - "voltage of solid state power amplifier bee"
vs_CPA1BV - "solid state power amplifier bee's voltage"
vs_CPA1BV - "Show me sea pea eh one bee vee"
vt_CPA1BV - "Tell me sea pea eh one bee vee"
vw_CPA1BV - "What is sea pea eh one bee vee"
vs_CPA1BV - "Show me the value of sea pea eh one bee vee"
vt_CPA1BV - "Tell me the value of sea pea eh one bee vee"
vw_CPA1BV - "What is the value of sea pea eh one bee vee"
vs_CPA1BV - "Show me the voltage of solid state power amplifier bee"
vt_CPA1BV - "Tell me the voltage of solid state power amplifier bee"
vw_CPA1BV - "What is the voltage of solid state power amplifier bee"
vs_CPA1BV - "Show me solid state power amplifier bee's voltage"
vw_CPA1BV - "Tell me solid state power amplifier bee's voltage"
vs_CPA1BV - "What is solid state power amplifier bee's voltage"

<u>CPA1BW</u>
vw_CPA1BW - "sea pea eh one bee double you"
vw_CPA1BW - "are eff power of solid state amplifier bee"
vw_CPA1BW - "solid state amplifier bee's are eff power"
vs_CPA1BW - "Show me sea pea eh one bee double you"
vt_CPA1BW - "Tell me sea pea eh one bee double you"
vw_CPA1BW - "What is sea pea eh one bee double you"
vs_CPA1BW - "Show me the value of sea pea eh one bee double you"
vt_CPA1BW - "Tell me the value of sea pea eh one bee double you"
vw_CPA1BW - "What is the value of sea pea eh one bee double you"
vs_CPA1BW - "Show me the are eff power of solid state amplifier bee"
vt_CPA1BW - "Tell me the are eff power of solid state amplifier bee"
vw_CPA1BW - "What is the are eff power of solid state amplifier bee"
vs_CPA1BW - "Show me solid state amplifier bee's are eff power"
vt_CPA1BW - "Tell me solid state amplifier bee's are eff power"
vw_CPA1BW - "What is solid state amplifier bee's are eff power"

<u>CT1AMT</u>
vw_CT1AMT - "sea tea one eh em tea"
vw_CT1AMT - "temperature of transmitter eh"
vw_CT1AMT - "transmitter eh's temperature"
vs_CT1AMT - "Show me sea tea one eh em tea"
vt_CT1AMT - "Tell me sea tea one eh em tea"
vw_CT1AMT - "What is sea tea one eh em tea"
vs_CT1AMT - "Show me the value of sea tea one eh em tea"
vt_CT1AMT - "Tell me the value of sea tea one eh em tea"
vw_CT1AMT - "What is the value of sea tea one eh em tea"
vs_CT1AMT - "Show me the temperature of transmitter eh"
vt_CT1AMT - "Tell me the temperature of transmitter eh"
vw_CT1AMT - "What is the temperature of transmitter eh"
vs_CT1AMT - "Show me transmitter eh's temperature"
vt_CT1AMT - "Tell me transmitter eh's temperature"
vw_CT1AMT - "What is transmitter eh's temperature"

CT1BMT

vw_CT1BMT - "sea tea one bee em tea"
vw_CT1BMT - "temperature of transmitter bee"
vw_CT1BMT - "transmitter bee's temperature"
vs_CT1BMT - "Show me sea tea one bee em tea"
vt_CT1BMT - "Tell me sea tea one bee em tea"
vw_CT1BMT - "What is sea tea one bee em tea"
vs_CT1BMT - "Show me the value of sea tea one bee em tea"
vt_CT1BMT - "Tell me the value of sea tea one bee em tea"
vw_CT1BMT - "What is the value of sea tea one bee em tea"
vs_CT1BMT - "Show me the temperature of transmitter bee"
vt_CT1BMT - "Tell me the temperature of transmitter bee"
vw_CT1BMT - "What is the temperature of transmitter bee"
vs_CT1BMT - "Show me transmitter bee's temperature"
vt_CT1BMT - "Tell me transmitter bee's temperature"
vw_CT1BMT - "What is transmitter bee's temperature"

PROPULSION SUBSYSTEM

GGPRES

vw_GGPRES - "gee gee pea are ee ess"
vw_GGPRES - "gee gee prez"
vw_GGPRES - "gas generator pressure"
vs_GGPRES - "pressure of the gas generator"
vs_GGPRES - "Show me gee gee pea are ee ess"
vt_GGPRES - "Tell me gee gee pea are ee ess"
vw_GGPRES - "What is gee gee pea are ee ess"
vs_GGPRES - "Show me the value of gee gee pea are ee ess"
vt_GGPRES - "Tell me the value of gee gee pea are ee ess"
vw_GGPRES - "What is the value of gee gee pea are ee ess"
vs_GGPRES - "Show me gee gee prez"
vt_GGPRES - "Tell me gee gee prez"

vw_GGPRES - "What is gee gee prez"
vs_GGPRES - "Show me the value of gee gee prez"
vt_GGPRES - "Tell me the value of gee gee prez"
vw_GGPRES - "What is the value of gee gee prez"
vs_GGPRES - "Show me the gas generator pressure"
vt_GGPRES - "Tell me the gas generator pressure"
vw_GGPRES - "What is the gas generator pressure"
vs_GGPRES - "Show me the pressure of the gas generator"
vt_GGPRES - "Tell me the pressure of the gas generator"
vw_GGPRES - "What is the pressure of the gas generator"

LVLSEL
vw_LVLSEL - "ell vee ell ess ee ell"
vw_LVLSEL - "level selected for plenum pressure"
vw_LVLSEL - "plenum pressure level selected"
vs_LVLSEL - "Show me ell vee ell ess ee ell"
vt_LVLSEL - "Tell me ell vee ell ess ee ell"
vw_LVLSEL - "What is ell vee ell ess ee ell"
vs_LVLSEL - "Show me the value of ell vee ell ess ee ell"
vt_LVLSEL - "Tell me the value of ell vee ell ess ee ell"
vw_LVLSEL - "What is the value of ell vee ell ess ee ell"
vs_LVLSEL - "Show me the level selected for plenum pressure"
vt_LVLSEL - "Tell me the level selected for plenum pressure"
vw_LVLSEL - "What is the level selected for plenum pressure"
vs_LVLSEL - "Show me the plenum pressure level selected"
vt_LVLSEL - "Tell me the plenum pressure level selected"
vw_LVLSEL - "What is the plenum pressure level selected"

P+XFBT
vw_P_PLUS_XFBT - "pea plus ex eff bee tea"
vw_P_PLUS_XFBT - "propellant filter body temperature"
vs_P_PLUS_XFBT - "temperature of the propellant filter body"
vs_P_PLUS_XFBT - "Show me pea plus ex eff bee tea"
vt_P_PLUS_XFBT - "Tell me pea plus ex eff bee tea"

vw_P_PLUS_XFBT - "What is pea plus ex eff bee tea"
vs_P_PLUS_XFBT - "Show me the value of pea plus ex eff bee tea"
vt_P_PLUS_XFBT - "Tell me the value of pea plus ex eff bee tea"
vw_P_PLUS_XFBT - "What is the value of pea plus ex eff bee tea"
vs_P_PLUS_XFBT - "Show me the propellant filter body temperature"
vt_P_PLUS_XFBT - "Tell me the propellant filter body temperature"
vw_P_PLUS_XFBT - "What is the propellant filter body temperature"
vs_P_PLUS_XFBT - "Show me the temperature of the propellant filter body"
vt_P_PLUS_XFBT - "Tell me the temperature of the propellant filter body"
vs_P_PLUS_XFBT - "What is the temperature of the propellant filter body"

P+XPTP
vw_P_PLUS_XPTP - "pea plus ex pea tea pea"
vw_P_PLUS_XPTP - "propellant tank pressure"
vw_P_PLUS_XPTP - "pressure of the propellant tank"
vs_P_PLUS_XPTP - "Show me pea plus ex pea tea pea"
vt_P_PLUS_XPTP - "Tell me pea plus ex pea tea pea"
vw_P_PLUS_XPTP - "What is pea plus ex pea tea pea"
vs_P_PLUS_XPTP - "Show me the value of pea plus ex pea tea pea"
vt_P_PLUS_XPTP - "Tell me the value of pea plus ex pea tea pea"
vw_P_PLUS_XPTP - "What is the value of pea plus ex pea tea pea"
vs_P_PLUS_XPTP - "Show me the propellant tank pressure"
vt_P_PLUS_XPTP - "Tell me the propellant tank pressure"
vw_P_PLUS_XPTP - "What is the propellant tank pressure"
vs_P_PLUS_XPTP - "Show me the pressure of the propellant tank"
vt_P_PLUS_XPTP - "Tell me the pressure of the propellant tank"
vw_P_PLUS_XPTP - "What is the pressure of the propellant tank"

P+XTOT
vw_P_PLUS_XTOT - "What is pea plus ex tea oh tea"
vw_P_PLUS_XTOT - "What is pea plus ex tot"
vw_P_PLUS_XTOT - "outboard propellant tank temperature"
vw_P_PLUS_XTOT - "temperature of the outboard propellant tank"
vs_P_PLUS_XTOT - "Show me pea plus ex tea oh tea"

vt_P_PLUS_XTOT - "Tell me pea plus ex tea oh tea"
vw_P_PLUS_XTOT - "What is pea plus ex tea oh tea"
vs_P_PLUS_XTOT - "Show me the value of pea plus ex tea oh tea"
vt_P_PLUS_XTOT - "Tell me the value of pea plus ex tea oh tea"
vw_P_PLUS_XTOT - "What is the value of pea plus ex tea oh tea"
vs_P_PLUS_XTOT - "Show me pea plus ex tot"
vt_P_PLUS_XTOT - "Tell me pea plus ex tot"
vw_P_PLUS_XTOT - "What is pea plus ex tot"
vs_P_PLUS_XTOT - "Show me the value of pea plus ex tot"
vt_P_PLUS_XTOT - "Tell me the value of pea plus ex tot"
vw_P_PLUS_XTOT - "What is the value of pea plus ex tot"
vs_P_PLUS_XTOT - "Show me the outboard propellant tank temperature"
vt_P_PLUS_XTOT - "Tell me the outboard propellant tank temperature"
vw_P_PLUS_XTOT - "What is the outboard propellant tank temperature"
vs_P_PLUS_XTOT - "Show me the temperature of the outboard propellant tank"
vt_P_PLUS_XTOT - "Tell me the temperature of the outboard propellant tank"
vw_P_PLUS_XTOT - "What is the temperature of the outboard propellant tank"

315

P+XVBT
vw_P_PLUS_XVBT - "pea plus ex vee bee tea"
vw_P_PLUS_XVBT - "propellant eye ess oh valve temperature"
vw_P_PLUS_XVBT - "temperature of the propellant eye ess oh valve"
vs_P_PLUS_XVBT - "Show me pea plus ex vee bee tea"
vt_P_PLUS_XVBT - "Tell me pea plus ex vee bee tea"
vw_P_PLUS_XVBT - "What is pea plus ex vee bee tea"
vs_P_PLUS_XVBT - "Show me the value of pea plus ex vee bee tea"
vt_P_PLUS_XVBT - "Tell me the value of pea plus ex vee bee tea"
vw_P_PLUS_XVBT - "What is the value of pea plus ex vee bee tea"
vs_P_PLUS_XVBT - "Show me the propellant eye ess oh valve temperature"
vt_P_PLUS_XVBT - "Tell me the propellant eye ess oh valve temperature"
vw_P_PLUS_XVBT - "What is the propellant eye ess oh valve temperature"
vs_P_PLUS_XVBT - "Show me the temperature of the propellant eye ess oh valve"
vt_P_PLUS_XVBT - "Tell me the temperature of the propellant eye ess oh valve"
vw_P_PLUS_XVBT - "What is the temperature of the propellant eye ess oh valve"

PFDIVT

vw_PFDIVT - "pea eff dee eye vee tea"
vw_PFDIVT - "propellant fill drain valve temperature"
vw_PFDIVT - "temperature of the propellant fill drain valve"
vs_PFDIVT - "Show me pea eff dee eye vee tea"
vt_PFDIVT - "Tell me pea eff dee eye vee tea"
vw_PFDIVT - "What is pea eff dee eye vee tea"
vs_PFDIVT - "Show me the value of pea eff dee eye vee tea"
vt_PFDIVT - "Tell me the value of pea eff dee eye vee tea"
vw_PFDIVT - "What is the value of pea eff dee eye vee tea"
vs_PFDIVT - "Show me the propellant fill drain valve temperature"
vt_PFDIVT - "Tell me the propellant fill drain valve temperature"
vw_PFDIVT - "What is the propellant fill drain valve temperature"
vs_PFDIVT - "Show me the temperature of the propellant fill drain valve"
vt_PFDIVT - "Tell me temperature of the propellant fill drain valve"
vw_PFDIVT - "What is temperature of the propellant fill drain valve"

PGGFPT

vw_PGGFPT - "pea gee gee eff pea tea"
vw_PGGFPT - "gas generator temperature"
vw_PGGFPT - "temperature of the gas generator"
vs_PGGFPT - "Show me pea gee gee eff pea tea"
vt_PGGFPT - "Tell me pea gee gee eff pea tea"
vw_PGGFPT - "What is pea gee gee eff pea tea"
vs_PGGFPT - "Show me the value of pea gee gee eff pea tea"
vt_PGGFPT - "Tell me the value of pea gee gee eff pea tea"
vw_PGGFPT - "What is the value of pea gee gee eff pea tea"
vs_PGGFPT - "Show me the gas generator temperature"
vt_PGGFPT - "Tell me the gas generator temperature"
vw_PGGFPT - "What is the gas generator temperature"
vs_PGGFPT - "Show me the temperature of the gas generator"
vt_PGGFPT - "Tell me the temperature of the gas generator"
vw_PGGFPT - "What is the temperature of the gas generator"

PHLT+T

vw_PHLT_PLUS_T - "pea H ell tea plus tea"
vw_PHLT_PLUS_T - "temperature of the plus ex high level thruster propellant"
vw_PHLT_PLUS_T - "plus ex high level thruster propellant temperature"
vs_PHLT_PLUS_T - "Show me pea H ell tea plus tea"
vt_PHLT_PLUS_T - "Tell me pea H ell tea plus tea"
vw_PHLT_PLUS_T - "What is pea H ell tea plus tea"
vs_PHLT_PLUS_T - "Show me the value of pea H ell tea plus tea"
vt_PHLT_PLUS_T - "Tell me the value of pea H ell tea plus tea"
vw_PHLT_PLUS_T - "What is the value of pea H ell tea plus tea"
vs_PHLT_PLUS_T - "Show me the temperature of the plus ex high level thruster propellant"
vt_PHLT_PLUS_T - "Tell me the temperature of the plus ex high level thruster propellant"
vw_PHLT_PLUS_T - "What is the temperature of the plus ex high level thruster propellant"
vs_PHLT_PLUS_T - "Show me the plus ex high level thruster propellant temperature"
vt_PHLT_PLUS_T - "Tell me the plus ex high level thruster propellant temperature"
vw_PHLT_PLUS_T - "What is the plus ex high level thruster propellant temperature"

PHLT-T

vw_PHLT_MINUS_T - "pea H ell tea minus tea"
vw_PHLT_MINUS_T - "temperature of the minus ex high level thruster propellant"
vw_PHLT_MINUS_T - "minus ex high level thruster propellant temperature"
vs_PHLT_MINUS_T - "Show me pea H ell tea minus tea"
vt_PHLT_MINUS_T - "Tell me pea H ell tea minus tea"
vw_PHLT_MINUS_T - "What is pea H ell tea minus tea"
vs_PHLT_MINUS_T - "Show me the value of pea H ell tea minus tea"
vt_PHLT_MINUS_T - "Tell me the value of pea H ell tea minus tea"
vw_PHLT_MINUS_T - "What is the value of pea H ell tea minus tea"
vs_PHLT_MINUS_T - "Show me the temperature of the minus ex high level thruster propellant"
vt_PHLT_MINUS_T - "Tell me the temperature of the minus ex high level thruster propellant"
vw_PHLT_MINUS_T - "What is the temperature of the minus ex high level thruster propellant"
vs_PHLT_MINUS_T - "Show me the minus ex high level thruster propellant temperature"
vt_PHLT_MINUS_T - "Tell me the minus ex high level thruster propellant temperature"
vw_PHLT_MINUS_T - "What is the minus ex high level thruster propellant temperature"

PHT+HT

vw_PHT_PLUS_HT - "pea H tea plus H tea"
vw_PHT_PLUS_HT - "temperature of the plus ex hydrazine line high level thruster"
vw_PHT_PLUS_HT - "plus ex hydrazine line high level thruster temperature"
vs_PHT_PLUS_HT - "Show me pea H tea plus H tea"
vt_PHT_PLUS_HT - "Tell me pea H tea plus H tea"
vw_PHT_PLUS_HT - "What is pea H tea plus H tea"
vs_PHT_PLUS_HT - "Show me the value of pea H tea plus H tea"
vt_PHT_PLUS_HT - "Tell me the value of pea H tea plus H tea"
vw_PHT_PLUS_HT - "What is the value of pea H tea plus H tea"
vs_PHT_PLUS_HT - "Show me the temperature of the plus ex hydrazine line high level thruster"
vt_PHT_PLUS_HT - "Tell me the temperature of the plus ex hydrazine line high level thruster"
vw_PHT_PLUS_HT - "What is the temperature of the plus ex hydrazine line high level thruster"
vs_PHT_PLUS_HT - "Show me the plus ex hydrazine line high level thruster temperature"
vt_PHT_PLUS_HT - "Tell me the plus ex hydrazine line high level thruster temperature"
vw_PHT_PLUS_HT - "What is the plus ex hydrazine line high level thruster temperature"

PHT-HT

vw_PHT_MINUS_HT - "pea H tea minus H tea"
vw_PHT_MINUS_HT - "temperature of the minus ex hydrazine line high level thruster"
vw_PHT_MINUS_HT - "minus ex hydrazine line high level thruster temperature"
vs_PHT_MINUS_HT - "Show me pea H tea minus H tea"
vt_PHT_MINUS_HT - "Tell me pea H tea minus H tea"
vw_PHT_MINUS_HT - "What is pea H tea minus H tea"
vs_PHT_MINUS_HT - "Show me the value of pea H tea minus H tea"
vt_PHT_MINUS_HT - "Tell me the value of pea H tea minus H tea"
vw_PHT_MINUS_HT - "What is the value of pea H tea minus H tea"
vs_PHT_MINUS_HT - "Show me the temperature of the minus ex hydrazine line high level thruster"
vt_PHT_MINUS_HT - "Tell me the temperature of the minus ex hydrazine line high level thruster"
vw_PHT_MINUS_HT - "What is the temperature of the minus ex hydrazine line high level thruster"
vs_PHT_MINUS_HT - "Show me the minus ex hydrazine line high level thruster temperature"
vt_PHT_MINUS_HT - "Tell me the minus ex hydrazine line high level thruster temperature"
vw_PHT_MINUS_HT - "What is the minus ex hydrazine line high level thruster temperature"

PPLN1P

vw_PPLN1P - "pea pea ell en one pea"
vw_PPLN1P - "plenum one pressure"
vw_PPLN1P - "pressure in plenum one"
vs_PPLN1P - "Show me pea pea ell en one pea"
vt_PPLN1P - "Tell me pea pea ell en one pea"
vw_PPLN1P - "What is pea pea ell en one pea"
vs_PPLN1P - "Show me the value of pea pea ell en one pea"
vt_PPLN1P - "Tell me the value of pea pea ell en one pea"
vw_PPLN1P - "What is the value of pea pea ell en one pea"
vs_PPLN1P - "Show me the plenum one pressure"
vt_PPLN1P - "Tell me the plenum one pressure"
vw_PPLN1P - "What is the plenum one pressure"
vs_PPLN1P - "Show me the pressure in plenum one"
vt_PPLN1P - "Tell me the pressure in plenum one"
vw_PPLN1P - "What is the pressure in plenum one"

PPLN2P

vw_PPLN2P - "pea pea ell en two pea"
vw_PPLN2P - "plenum two pressure"
vw_PPLN2P - "pressure in plenum two"
vs_PPLN2P - "Show me pea pea ell en two pea"
vt_PPLN2P - "Tell me pea pea ell en two pea"
vw_PPLN2P - "What is pea pea ell en two pea"
vs_PPLN2P - "Show me the value of pea pea ell en two pea"
vt_PPLN2P - "Tell me the value of pea pea ell en two pea"
vw_PPLN2P - "What is the value of pea pea ell en two pea"
vs_PPLN2P - "Show me the plenum two pressure"
vt_PPLN2P - "Tell me the plenum two pressure"
vw_PPLN2P - "What is the plenum two pressure"
vs_PPLN2P - "Show me the pressure in plenum two"
vt_PPLN2P - "Tell me the pressure in plenum two"
vw_PPLN2P - "What is the pressure in plenum two"

<u>PVT+HT</u>

vw_PVT_PLUS_HT - "pea vee tea plus H tea"
vw_PVT_PLUS_HT - "temperature of the plus ex hydrazine line delta vee thruster"
vw_PVT_PLUS_HT - "plus ex hydrazine line delta vee thruster temperature"
vs_PVT_PLUS_HT - "Show me pea vee tea plus H tea"
vt_PVT_PLUS_HT - "Tell me pea vee tea plus H tea"
vw_PVT_PLUS_HT - "What is pea vee tea plus H tea"
vs_PVT_PLUS_HT - "Show me the value of pea vee tea plus H tea"
vt_PVT_PLUS_HT - "Tell me the value of pea vee tea plus H tea"
vw_PVT_PLUS_HT - "What is the value of pea vee tea plus H tea"
vs_PVT_PLUS_HT - "Show me the temperature of the plus ex hydrazine line delta vee thruster"
vt_PVT_PLUS_HT - "Tell me the temperature of the plus ex hydrazine line delta vee thruster"
vw_PVT_PLUS_HT - "What is the temperature of the plus ex hydrazine line delta vee thruster"
vs_PVT_PLUS_HT - "Show me the plus ex hydrazine line delta vee thruster temperature"
vt_PVT_PLUS_HT - "Tell me the plus ex hydrazine line delta vee thruster temperature"
vw_PVT_PLUS_HT - "What is the plus ex hydrazine line delta vee thruster temperature"

<u>PVT-HT</u>

vw_PVT_MINUS_HT - "pea vee tea minus H tea"
vw_PVT_MINUS_HT - "temperature of the minus ex hydrazine line delta vee thruster"
vw_PVT_MINUS_HT - "minus ex hydrazine line delta vee thruster temperature"
vs_PVT_MINUS_HT - "Show me pea vee tea minus H tea"
vt_PVT_MINUS_HT - "Tell me pea vee tea minus H tea"
vw_PVT_MINUS_HT - "What is pea vee tea minus H tea"
vs_PVT_MINUS_HT - "Show me the value of pea vee tea minus H tea"
vt_PVT_MINUS_HT - "Tell me the value of pea vee tea minus H tea"
vw_PVT_MINUS_HT - "What is the value of pea vee tea minus H tea"
vs_PVT_MINUS_HT - "Show me the temperature of the minus ex hydrazine line delta vee thruster"
vt_PVT_MINUS_HT - "Tell me the temperature of the minus ex hydrazine line delta vee thruster"
vw_PVT_MINUS_HT - "What is the temperature of the minus ex hydrazine line delta vee thruster"
vs_PVT_MINUS_HT - "Show me the minus ex hydrazine line delta vee thruster temperature"
vt_PVT_MINUS_HT - "Tell me the minus ex hydrazine line delta vee thruster temperature"
vw_PVT_MINUS_HT - "What is the minus ex hydrazine line delta vee thruster temperature"

PVVT+T

vw_PVVT_PLUS_T - "pea vee vee tea plus tea"
vw_PVVT_PLUS_T - "temperature of the plus ex delta vee thruster propellant"
vw_PVVT_PLUS_T - "plus ex delta vee thruster propellant temperature"
vs_PVVT_PLUS_T - "Show me pea vee vee tea plus tea"
vt_PVVT_PLUS_T - "Tell me pea vee vee tea plus tea"
vw_PVVT_PLUS_T - "What is pea vee vee tea plus tea"
vs_PVVT_PLUS_T - "Show me the value of pea vee vee tea plus tea"
vt_PVVT_PLUS_T - "Tell me the value of pea vee vee tea plus tea"
vw_PVVT_PLUS_T - "What is the value of pea vee vee tea plus tea"
vs_PVVT_PLUS_T - "Show me the temperature of the plus ex delta vee thruster propellant"
vt_PVVT_PLUS_T - "Tell me the temperature of the plus ex delta vee thruster propellant"
vw_PVVT_PLUS_T - "What is the temperature of the plus ex delta vee thruster propellant"
vs_PVVT_PLUS_T - "Show me the plus ex delta vee thruster propellant temperature"
vt_PVVT_PLUS_T - "Tell me the plus ex delta vee thruster propellant temperature"
vw_PVVT_PLUS_T - "What is the plus ex delta vee thruster propellant temperature"

PVVT-T

vw_PVVT_MINUS_T - "pea vee vee tea minus tea"
vw_PVVT_MINUS_T - "temperature of the minus ex delta vee thruster propellant"
vw_PVVT_MINUS_T - "minus ex delta vee thruster propellant temperature"
vs_PVVT_MINUS_T - "Show me pea vee vee tea minus tea"
vt_PVVT_MINUS_T - "Tell me pea vee vee tea minus tea"
vw_PVVT_MINUS_T - "What is pea vee vee tea minus tea"
vs_PVVT_MINUS_T - "Show me the value of pea vee vee tea minus tea"
vt_PVVT_MINUS_T - "Tell me the value of pea vee vee tea minus tea"
vw_PVVT_MINUS_T - "What is the value of pea vee vee tea minus tea"
vs_PVVT_MINUS_T - "Show me the temperature of the minus ex delta vee thruster propellant"
vt_PVVT_MINUS_T - "Tell me the temperature of the minus ex delta vee thruster propellant"
vw_PVVT_MINUS_T - "What is the temperature of the minus ex delta vee thruster propellant"
vs_PVVT_MINUS_T - "Show me the minus ex delta vee thruster propellant temperature"
vt_PVVT_MINUS_T - "Tell me the minus ex delta vee thruster propellant temperature"
vw_PVVT_MINUS_T - "What is the minus ex delta vee thruster propellant temperature"

P-XCLT

vw_P_MINUS_XCLT - "pea minus ex sea ell tea"
vw_P_MINUS_XCLT - "temperature of the propellant crossover line"
vw_P_MINUS_XCLT - "propellant crossover line temperature"
vs_P_MINUS_XCLT - "Show me pea minus ex sea ell tea"
vt_P_MINUS_XCLT - "Tell me pea minus ex sea ell tea"
vw_P_MINUS_XCLT - "What is pea minus ex sea ell tea"
vs_P_MINUS_XCLT - "Show me the value of pea minus ex sea ell tea"
vt_P_MINUS_XCLT - "Tell me the value of pea minus ex sea ell tea"
vw_P_MINUS_XCLT - "What is the value of pea minus ex sea ell tea"
vs_P_MINUS_XCLT - "Show me the temperature of the propellant crossover line"
vt_P_MINUS_XCLT - "Tell me the temperature of the propellant crossover line"
vw_P_MINUS_XCLT - "What is the temperature of the propellant crossover line"
vs_P_MINUS_XCLT - "Show me the propellant crossover line temperature"
vt_P_MINUS_XCLT - "Tell me the propellant crossover line temperature"
vw_P_MINUS_XCLT - "What is the propellant crossover line temperature"

P-XFBT

vw_P_MINUS_XFBT - "pea minus ex eff bee tea"
vw_P_MINUS_XFBT - "temperature of the propellant ex filter body"
vw_P_MINUS_XFBT - "propellant ex filter body temperature"
vs_P_MINUS_XFBT - "Show me pea minus ex eff bee tea"
vt_P_MINUS_XFBT - "Tell me pea minus ex eff bee tea"
vw_P_MINUS_XFBT - "What is pea minus ex eff bee tea"
vs_P_MINUS_XFBT - "Show me the value of pea minus ex eff bee tea"
vt_P_MINUS_XFBT - "Tell me the value of pea minus ex eff bee tea"
vw_P_MINUS_XFBT - "What is the value of pea minus ex eff bee tea"
vs_P_MINUS_XFBT - "Show me the temperature of the propellant ex filter body"
vt_P_MINUS_XFBT - "Tell me the temperature of the propellant ex filter body"
vw_P_MINUS_XFBT - "What is the temperature of the propellant ex filter body"
vs_P_MINUS_XFBT - "Show me the propellant ex filter body temperature"
vt_P_MINUS_XFBT - "Tell me the propellant ex filter body temperature"
vw_P_MINUS_XFBT - "What is the propellant ex filter body temperature"

P-XPTP

vw_P_MINUS_XPTP - "pea minus ex pea tea pea"
vw_P_MINUS_XPTP - "pressure of the propellant ex tank"
vw_P_MINUS_XPTP - "propellant ex tank pressure"
vs_P_MINUS_XPTP - "Show me pea minus ex pea tea pea"
vt_P_MINUS_XPTP - "Tell me pea minus ex pea tea pea"
vw_P_MINUS_XPTP - "What is pea minus ex pea tea pea"
vs_P_MINUS_XPTP - "Show me the value of pea minus ex pea tea pea"
vt_P_MINUS_XPTP - "Tell me the value of pea minus ex pea tea pea"
vw_P_MINUS_XPTP - "What is the value of pea minus ex pea tea pea"
vs_P_MINUS_XPTP - "Show me the pressure of the propellant ex tank"
vt_P_MINUS_XPTP - "Tell me the pressure of the propellant ex tank"
vw_P_MINUS_XPTP - "What is the pressure of the propellant ex tank"
vs_P_MINUS_XPTP - "Show me the propellant ex tank pressure"
vt_P_MINUS_XPTP - "Tell me the propellant ex tank pressure"
vw_P_MINUS_XPTP - "What is the propellant ex tank pressure"

P-XTIT

vw_P_MINUS_XTIT - "pea minus ex tea eye tea"
vw_P_MINUS_XTIT - "pea minus ex tit"
vw_P_MINUS_XTIT - "temperature of the inboard propellant tank"
vw_P_MINUS_XTIT - "inboard propellant tank temperature"
vs_P_MINUS_XTIT - "Show me pea minus ex tea eye tea"
vt_P_MINUS_XTIT - "Tell me pea minus ex tea eye tea"
vw_P_MINUS_XTIT - "What is pea minus ex tea eye tea"
vs_P_MINUS_XTIT - "Show me the value of pea minus ex tea eye tea"
vt_P_MINUS_XTIT - "Tell me the value of pea minus ex tea eye tea"
vw_P_MINUS_XTIT - "What is the value of pea minus ex tea eye tea"
vs_P_MINUS_XTIT - "Show me pea minus ex tit"
vt_P_MINUS_XTIT - "Tell me pea minus ex tit"
vw_P_MINUS_XTIT - "What is pea minus ex tit"
vs_P_MINUS_XTIT - "Show me the value of pea minus ex tit"
vt_P_MINUS_XTIT - "Tell me the value of pea minus ex tit"

vw_P_MINUS_XTIT - "What is the value of pea minus ex tit"
vs_P_MINUS_XTIT - "Show me the temperature of the inboard propellant tank"
vt_P_MINUS_XTIT - "Tell me the temperature of the inboard propellant tank"
vw_P_MINUS_XTIT - "What is the temperature of the inboard propellant tank"
vs_P_MINUS_XTIT - "Show me the inboard propellant tank temperature"
vt_P_MINUS_XTIT - "Tell me the inboard propellant tank temperature"
vw_P_MINUS_XTIT - "What is the inboard propellant tank temperature"

<u>P-XTOT</u>
vw_P_MINUS_XTOT - "pea minus ex tea oh tea"
vw_P_MINUS_XTOT - "pea minus ex tot"
vw_P_MINUS_XTOT - "temperature of the outboard propellant tank"
vw_P_MINUS_XTOT - "outboard propellant tank temperature"
vs_P_MINUS_XTOT - "Show me pea minus ex tea oh tea"
vt_P_MINUS_XTOT - "Tell me pea minus ex tea oh tea"
vw_P_MINUS_XTOT - "What is pea minus ex tea oh tea"
vs_P_MINUS_XTOT - "Show me the value of pea minus ex tea oh tea"
vt_P_MINUS_XTOT - "Tell me the value of pea minus ex tea oh tea"
vw_P_MINUS_XTOT - "What is the value of pea minus ex tea oh tea"
vs_P_MINUS_XTOT - "Show me pea minus ex tot"
vt_P_MINUS_XTOT - "Tell me pea minus ex tot"
vw_P_MINUS_XTOT - "What is pea minus ex tot"
vs_P_MINUS_XTOT - "Show me the value of pea minus ex tot"
vt_P_MINUS_XTOT - "Tell me the value of pea minus ex tot"
vw_P_MINUS_XTOT - "What is the value of pea minus ex tot"
vs_P_MINUS_XTOT - "Show me the temperature of the outboard propellant tank"
vt_P_MINUS_XTOT - "Tell me the temperature of the outboard propellant tank"
vw_P_MINUS_XTOT - "What is the temperature of the outboard propellant tank"
vs_P_MINUS_XTOT - "Show me the outboard propellant tank temperature"
vt_P_MINUS_XTOT - "Tell me the outboard propellant tank temperature"
vw_P_MINUS_XTOT - "What is the outboard propellant tank temperature"

<u>P-XVBT</u>
vw_P_MINUS_XVBT - "pea minus ex vee bee tea"

vw_P_MINUS_XVBT - "temperature of the propellant thruster eye ess oh valve"
vw_P_MINUS_XVBT - "propellant thruster eye ess oh valve temperature"
vs_P_MINUS_XVBT - "Show me pea minus ex vee bee tea"
vt_P_MINUS_XVBT - "Tell me pea minus ex vee bee tea"
vw_P_MINUS_XVBT - "What is pea minus ex vee bee tea"
vs_P_MINUS_XVBT - "Show me the value of pea minus ex vee bee tea"
vt_P_MINUS_XVBT - "Tell me the value of pea minus ex vee bee tea"
vw_P_MINUS_XVBT - "What is the value of pea minus ex vee bee tea"
vs_P_MINUS_XVBT - "Show me the temperature of the propellant thruster eye ess oh valve"
vt_P_MINUS_XVBT - "Tell me the temperature of the propellant thruster eye ess oh valve"
vw_P_MINUS_XVBT - "What is the temperature of the propellant thruster eye ess oh valve"
vs_P_MINUS_XVBT - "Show me the propellant thruster eye ess oh valve temperature"
vt_P_MINUS_XVBT - "Tell me the propellant thruster eye ess oh valve temperature"
vw_P_MINUS_XVBT - "What is the propellant thruster eye ess oh valve temperature"

ELECTRICAL POWER SUBSYSTEM

ECUSBI

vw_ECUSBI - "ee sea you ess bee eye"
vw_ECUSBI - "shunt bus current"
vw_ECUSBI - "current through the shunt bus"
vs_ECUSBI - "Show me ee sea you ess bee eye"
vt_ECUSBI - "Tell me ee sea you ess bee eye"
vw_ECUSBI - "What is ee sea you ess bee eye"
vs_ECUSBI - "Show me the value of ee sea you ess bee eye"
vt_ECUSBI - "Tell me the value of ee sea you ess bee eye"
vw_ECUSBI - "What is the value of ee sea you ess bee eye"
vs_ECUSBI - "Show me the shunt bus current"
vt_ECUSBI - "Tell me the shunt bus current"
vw_ECUSBI - "What is the shunt bus current"
vs_ECUSBI - "Show me the current through the shunt bus"
vt_ECUSBI - "Tell me the current through the shunt bus"

vw_ECUSBI - "What is the current through the shunt bus"

ECUSBV

vw_ECUSBV - "ee sea you ess bee vee"
vw_ECUSBV - "shunt bus voltage"
vs_ECUSBV - "voltage at the shunt bus"
vs_ECUSBV - "Show me ee sea you ess bee vee"
vt_ECUSBV - "Tell me ee sea you ess bee vee"
vw_ECUSBV - "What is ee sea you ess bee vee"
vs_ECUSBV - "Show me the value of ee sea you ess bee vee"
vt_ECUSBV - "Tell me the value of ee sea you ess bee vee"
vw_ECUSBV - "What is the value of ee sea you ess bee vee"
vs_ECUSBV - "Show me the shunt bus voltage"
vt_ECUSBV - "Tell me the shunt bus voltage"
vw_ECUSBV - "What is the shunt bus voltage"
vs_ECUSBV - "Show me the voltage at the shunt bus"
vt_ECUSBV - "Tell me the voltage at the shunt bus"
vs_ECUSBV - "What is the voltage at the shunt bus"

EED1AV

vw_EED1AV - "ee ee dee one eh vee"
vw_EED1AV - "output voltage at ten volt electrical distribution unit converter eh"
vw_EED1AV - "ten volt electrical distribution unit converter eh's output voltage"
vs_EED1AV - "Show me ee ee dee one eh vee"
vt_EED1AV - "Tell me ee ee dee one eh vee"
vw_EED1AV - "What is ee ee dee one eh vee"
vs_EED1AV - "Show me the value of ee ee dee one eh vee"
vt_EED1AV - "Tell me the value of ee ee dee one eh vee"
vw_EED1AV - "What is the value of ee ee dee one eh vee"
vs_EED1AV - "Show me the output voltage at ten volt electrical distribution unit converter eh"
vt_EED1AV - "Tell me the output voltage at ten volt electrical distribution unit converter eh"
vw_EED1AV - "What is the output voltage at ten volt electrical distribution unit converter eh"
vs_EED1AV - "Show me the ten volt electrical distribution unit converter eh's output voltage"
vt_EED1AV - "Tell me the ten volt electrical distribution unit converter eh's output voltage"

vw_EED1AV - "What is the ten volt electrical distribution unit converter eh's output voltage"

<u>EED1BV</u>
vw_EED1BV - "ee ee dee one bee vee"
vw_EED1BV - "output voltage at ten volt electrical distribution unit converter bee"
vw_EED1BV - "ten volt electrical distribution unit converter bee's output voltage"
vs_EED1BV - "Show me ee ee dee one bee vee"
vt_EED1BV - "Tell me ee ee dee one bee vee"
vw_EED1BV - "What is ee ee dee one bee vee"
vs_EED1BV - "Show me the value of ee ee dee one bee vee"
vt_EED1BV - "Tell me the value of ee ee dee one bee vee"
vw_EED1BV - "What is the value of ee ee dee one bee vee"
vs_EED1BV - "Show me the output voltage at ten volt electrical distribution unit converter bee"
vt_EED1BV - "Tell me the output voltage at ten volt electrical distribution unit converter bee"
vw_EED1BV - "What is the output voltage at ten volt electrical distribution unit converter bee"
vs_EED1BV - "Show me the ten volt electrical distribution unit converter bee's output voltage"
vt_EED1BV - "Tell me the ten volt electrical distribution unit converter bee's output voltage"
vw_EED1BV - "What is the ten volt electrical distribution unit converter bee's output voltage"

<u>EED5AV</u>
vw_EED5AV - "ee ee dee five eh vee"
vw_EED5AV - "output voltage at five volt electrical distribution unit converter eh"
vw_EED5AV - "five volt electrical distribution unit converter eh's output voltage"
vs_EED5AV - "Show me ee ee dee five eh vee"
vt_EED5AV - "Tell me ee ee dee five eh vee"
vw_EED5AV - "What is ee ee dee five eh vee"
vs_EED5AV - "Show me the value of ee ee dee five eh vee"
vt_EED5AV - "Tell me the value of ee ee dee five eh vee"
vw_EED5AV - "What is the value of ee ee dee five eh vee"
vs_EED5AV - "Show me the output voltage at five volt electrical distribution unit converter eh"
vt_EED5AV - "Tell me the output voltage at five volt electrical distribution unit converter eh"
vw_EED5AV - "What is the output voltage at five volt electrical distribution unit converter eh"
vs_EED5AV - "Show me the five volt electrical distribution unit converter eh's output voltage"
vt_EED5AV - "Tell me the five volt electrical distribution unit converter eh's output voltage"

vw_EED5AV - "What is the five volt electrical distribution unit converter eh's output voltage"

EED5BV

vw_EED5BV - "ee ee dee five bee vee"
vw_EED5BV - "output voltage at five volt electrical distribution unit converter bee"
vw_EED5BV - "five volt electrical distribution unit converter bee's output voltage"
vs_EED5BV - "Show me ee ee dee five bee vee"
vt_EED5BV - "Tell me ee ee dee five bee vee"
vw_EED5BV - "What is ee ee dee five bee vee"
vs_EED5BV - "Show me the value of ee ee dee five bee vee"
vt_EED5BV - "Tell me the value of ee ee dee five bee vee"
vw_EED5BV - "What is the value of ee ee dee five bee vee"
vs_EED5BV - "Show me the output voltage at five volt electrical distribution unit converter bee"
vt_EED5BV - "Tell me the output voltage at five volt electrical distribution unit converter bee"
vw_EED5BV - "What is the output voltage at five volt electrical distribution unit converter bee"
vs_EED5BV - "Show me the five volt electrical distribution unit converter bee's output voltage"
vt_EED5BV - "Tell me the five volt electrical distribution unit converter bee's output voltage"
vw_EED5BV - "What is the five volt electrical distribution unit converter bee's output voltage"

EPBA1T

vw_EPBA1T - "ee pea bee eh one tea"
vw_EPBA1T - "number one temperature of battery eh"
vw_EPBA1T - "battery eh's number one temperature"
vs_EPBA1T - "Show me ee pea bee eh one tea"
vt_EPBA1T - "Tell me ee pea bee eh one tea"
vw_EPBA1T - "What is ee pea bee eh one tea"
vs_EPBA1T - "Show me the value of ee pea bee eh one tea"
vt_EPBA1T - "Tell me the value of ee pea bee eh one tea"
vw_EPBA1T - "What is the value of ee pea bee eh one tea"
vs_EPBA1T - "Show me the number one temperature of battery eh"
vt_EPBA1T - "Tell me the number one temperature of battery eh"
vw_EPBA1T - "What is the number one temperature of battery eh"
vs_EPBA1T - "Show me battery eh's number one temperature"
vt_EPBA1T - "Tell me battery eh's number one temperature"

vw_EPBA1T - "What is battery eh's number one temperature"

EPBA2T

vw_EPBA2T - "ee pea bee eh two tea"
vw_EPBA2T - "number two temperature of battery eh"
vw_EPBA2T - "battery eh's number two temperature"
vs_EPBA2T - "Show me ee pea bee eh two tea"
vt_EPBA2T - "Tell me ee pea bee eh two tea"
vw_EPBA2T - "What is ee pea bee eh two tea"
vs_EPBA2T - "Show me the value of ee pea bee eh two tea"
vt_EPBA2T - "Tell me the value of ee pea bee eh two tea"
vw_EPBA2T - "What is the value of ee pea bee eh two tea"
vs_EPBA2T - "Show me the number two temperature of battery eh"
vt_EPBA2T - "Tell me the number two temperature of battery eh"
vw_EPBA2T - "What is the number two temperature of battery eh"
vs_EPBA2T - "Show me battery eh's number two temperature"
vt_EPBA2T - "Tell me battery eh's number two temperature"
vw_EPBA2T - "What is battery eh's number two temperature"

EPBB1T

vw_EPBB1T - "ee pea bee bee one tea"
vw_EPBB1T - "number one temperature of battery bee"
vw_EPBB1T - "battery bee's number one temperature"
vs_EPBB1T - "Show me ee pea bee bee one tea"
vt_EPBB1T - "Tell me ee pea bee bee one tea"
vw_EPBB1T - "What is ee pea bee bee one tea"
vs_EPBB1T - "Show me the value of ee pea bee bee one tea"
vt_EPBB1T - "Tell me the value of ee pea bee bee one tea"
vw_EPBB1T - "What is the value of ee pea bee bee one tea"
vs_EPBB1T - "Show me the number one temperature of battery bee"
vt_EPBB1T - "Tell me the number one temperature of battery bee"
vw_EPBB1T - "What is the number one temperature of battery bee"
vs_EPBB1T - "Show me battery bee's number one temperature"
vt_EPBB1T - "Tell me battery bee's number one temperature"

vw_EPBB1T - "What is battery bee's number one temperature"

EPBB2T

vw_EPBB2T - "ee pea bee bee two tea"
vw_EPBB2T - "number two temperature of battery bee"
vw_EPBB2T - "battery bee's number two temperature"
vs_EPBB2T - "Show me ee pea bee bee two tea"
vt_EPBB2T - "Tell me ee pea bee bee two tea"
vw_EPBB2T - "What is ee pea bee bee two tea"
vs_EPBB2T - "Show me the value of ee pea bee bee two tea"
vt_EPBB2T - "Tell me the value of ee pea bee bee two tea"
vw_EPBB2T - "What is the value of ee pea bee bee two tea"
vs_EPBB2T - "Show me the number two temperature of battery bee"
vt_EPBB2T - "Tell me the number two temperature of battery bee"
vw_EPBB2T - "What is the number two temperature of battery bee"
vs_EPBB2T - "Show me battery bee's number two temperature"
vt_EPBB2T - "Tell me battery bee's number two temperature"
vw_EPBB2T - "What is battery bee's number two temperature"

EPBC1T

vw_EPBC1T - "ee pea bee sea one tea"
vw_EPBC1T - "number one temperature of battery sea"
vw_EPBC1T - "battery sea's number one temperature"
vs_EPBC1T - "Show me ee pea bee sea one tea"
vt_EPBC1T - "Tell me ee pea bee sea one tea"
vw_EPBC1T - "What is ee pea bee sea one tea"
vs_EPBC1T - "Show me the value of ee pea bee sea one tea"
vt_EPBC1T - "Tell me the value of ee pea bee sea one tea"
vw_EPBC1T - "What is the value of ee pea bee sea one tea"
vs_EPBC1T - "Show me the number one temperature of battery sea"
vt_EPBC1T - "Tell me the number one temperature of battery sea"
vw_EPBC1T - "What is the number one temperature of battery sea"
vs_EPBC1T - "Show me battery sea's number one temperature"
vt_EPBC1T - "Tell me battery sea's number one temperature"

vw_EPBC1T - "What is battery sea's number one temperature"

EPBC2T
vw_EPBC2T - "ee pea bee sea two tea"
vw_EPBC2T - "number two temperature of battery sea"
vw_EPBC2T - "battery sea's number two temperature"
vs_EPBC2T - "Show me ee pea bee sea two tea"
vt_EPBC2T - "Tell me ee pea bee sea two tea"
vw_EPBC2T - "What is ee pea bee sea two tea"
vs_EPBC2T - "Show me the value of ee pea bee sea two tea"
vt_EPBC2T - "Tell me the value of ee pea bee sea two tea"
vw_EPBC2T - "What is the value of ee pea bee sea two tea"
vs_EPBC2T - "Show me the number two temperature of battery sea"
vt_EPBC2T - "Tell me the number two temperature of battery sea"
vw_EPBC2T - "What is the number two temperature of battery sea"
vs_EPBC2T - "Show me battery sea's number two temperature"
vt_EPBC2T - "Tell me battery sea's number two temperature"
vw_EPBC2T - "What is battery sea's number two temperature"

EPBSAI
vw_EPBSAI - "ee pea bee ess eh eye"
vw_EPBSAI - "primary solar bus current"
vw_EPBSAI - "current in the primary solar bus"
vs_EPBSAI - "Show me ee pea bee ess eh eye"
vt_EPBSAI - "Tell me ee pea bee ess eh eye"
vw_EPBSAI - "What is ee pea bee ess eh eye"
vs_EPBSAI - "Show me the value of ee pea bee ess eh eye"
vt_EPBSAI - "Tell me the value of ee pea bee ess eh eye"
vw_EPBSAI - "What is the value of ee pea bee ess eh eye"
vs_EPBSAI - "Show me the primary solar bus current"
vt_EPBSAI - "Tell me the primary solar bus current"
vw_EPBSAI - "What is the primary solar bus current"
vs_EPBSAI - "Show me the current in the primary solar bus"
vt_EPBSAI - "Tell me the current in the primary solar bus"

vw_EPBSAI - "What is the current in the primary solar bus"

<u>EPLP2T</u>

vw_EPLP2T - "ee pea ell pea two tea"
vw_EPLP2T - "solar array panel temperature"
vw_EPLP2T - "temperature of the solar array panel"
vs_EPLP2T - "Show me ee pea ell pea two tea"
vt_EPLP2T - "Tell me ee pea ell pea two tea"
vw_EPLP2T - "What is ee pea ell pea two tea"
vs_EPLP2T - "Show me the value of ee pea ell pea two tea"
vt_EPLP2T - "Tell me the value of ee pea ell pea two tea"
vw_EPLP2T - "What is the value of ee pea ell pea two tea"
vs_EPLP2T - "Show me the solar array panel temperature"
vt_EPLP2T - "Tell me the solar array panel temperature"
vw_EPLP2T - "What is the solar array panel temperature"
vs_EPLP2T - "Show me the temperature of the solar array panel"
vt_EPLP2T - "Tell me the temperature of the solar array panel"
vw_EPLP2T - "What is the temperature of the solar array panel"

<u>EPSBAI</u>

vw_EPSBAI - "ee pea ess bee eh eye"
vw_EPSBAI - "battery eh current"
vw_EPSBAI - "current through battery eh"
vs_EPSBAI - "Show me ee pea ess bee eh eye"
vt_EPSBAI - "Tell me ee pea ess bee eh eye"
vw_EPSBAI - "What is ee pea ess bee eh eye"
vs_EPSBAI - "Show me the value of ee pea ess bee eh eye"
vt_EPSBAI - "Tell me the value of ee pea ess bee eh eye"
vw_EPSBAI - "What is the value of ee pea ess bee eh eye"
vs_EPSBAI - "Show me the battery eh current"
vt_EPSBAI - "Tell me the battery eh current"
vw_EPSBAI - "What is the battery eh current"
vs_EPSBAI - "Show me the current through battery eh"
vt_EPSBAI - "Tell me the current through battery eh"

vw_EPSBAI - "What is the current through battery eh"

EPSBAV
vw_EPSBAV - "ee pea ess bee eh vee"
vw_EPSBAV - "battery eh's voltage"
vw_EPSBAV - "voltage in battery eh"
vs_EPSBAV - "Show me ee pea ess bee eh vee"
vt_EPSBAV - "Tell me ee pea ess bee eh vee"
vw_EPSBAV - "What is ee pea ess bee eh vee"
vs_EPSBAV - "Show me the value of ee pea ess bee eh vee"
vt_EPSBAV - "Tell me the value of ee pea ess bee eh vee"
vw_EPSBAV - "What is tha value of ee pea ess bee eh vee"
vs_EPSBAV - "Show me battery eh's voltage"
vt_EPSBAV - "Tell me battery eh's voltage"
vw_EPSBAV - "What is battery eh's voltage"
vs_EPSBAV - "Show me the voltage in battery eh"
vt_EPSBAV - "Tell me the voltage in battery eh"
vw_EPSBAV - "What is the voltage in battery eh"

333

EPSBBI
vw_EPSBBI - "ee pea ess bee bee eye"
vw_EPSBBI - "battery bee current"
vw_EPSBBI - "current through battery bee"
vs_EPSBBI - "Show me ee pea ess bee bee eye"
vt_EPSBBI - "Tell me ee pea ess bee bee eye"
vw_EPSBBI - "What is ee pea ess bee bee eye"
vs_EPSBBI - "Show me the value of ee pea ess bee bee eye"
vt_EPSBBI - "Tell me the value of ee pea ess bee bee eye"
vw_EPSBBI - "What is the value of ee pea ess bee bee eye"
vs_EPSBBI - "Show me the battery bee current"
vt_EPSBBI - "Tell me the battery bee current"
vw_EPSBBI - "What is the battery bee current"
vs_EPSBBI - "Show me the current through battery bee"
vt_EPSBBI - "Tell me the current through battery bee"

vw_EPSBBI - "What is the current through battery bee"

<u>EPSBBV</u>

vw_EPSBBV - "ee pea ess bee bee vee"
vw_EPSBBV - "battery bee's voltage"
vw_EPSBBV - "voltage in battery bee"
vs_EPSBBV - "Show me ee pea ess bee bee vee"
vt_EPSBBV - "Tell me ee pea ess bee bee vee"
vw_EPSBBV - "What is ee pea ess bee bee vee"
vs_EPSBBV - "Show me the value of ee pea ess bee bee vee"
vt_EPSBBV - "Tell me the value of ee pea ess bee bee vee"
vw_EPSBBV - "What is the value of ee pea ess bee bee vee"
vs_EPSBBV - "Show me battery bee's voltage"
vt_EPSBBV - "Tell me battery bee's voltage"
vw_EPSBBV - "What is battery bee's voltage"
vs_EPSBBV - "Show me the voltage in battery bee"
vt_EPSBBV - "Tell me the voltage in battery bee"
vw_EPSBBV - "What is the voltage in battery bee"

334

<u>EPSBCI</u>

vw_EPSBCI - "ee pea ess bee sea eye"
vw_EPSBCI - "battery sea current"
vw_EPSBCI - "current through battery sea"
vs_EPSBCI - "Show me ee pea ess bee sea eye"
vt_EPSBCI - "Tell me ee pea ess bee sea eye"
vw_EPSBCI - "What is ee pea ess bee sea eye"
vs_EPSBCI - "Show me the value of ee pea ess bee sea eye"
vt_EPSBCI - "Tell me the value of ee pea ess bee sea eye"
vw_EPSBCI - "What is the value of ee pea ess bee sea eye"
vs_EPSBCI - "Show me the battery sea current"
vt_EPSBCI - "Tell me the battery sea current"
vw_EPSBCI - "What is the battery sea current"
vs_EPSBCI - "Show me the current through battery sea"
vt_EPSBCI - "Tell me the current through battery sea"

vw_EPSBCI - "What is the current through battery sea"

EPSBCV

vw_EPSBCV - "ee pea ess bee sea vee"
vw_EPSBCV - "battery sea's voltage"
vw_EPSBCV - "voltage in battery sea"
vs_EPSBCV - "Show me ee pea ess bee sea vee"
vt_EPSBCV - "Tell me ee pea ess bee sea vee"
vw_EPSBCV - "What is ee pea ess bee sea vee"
vs_EPSBCV - "Show me the value of ee pea ess bee sea vee"
vt_EPSBCV - "Tell me the value of ee pea ess bee sea vee"
vw_EPSBCV - "What is the value of ee pea ess bee sea vee"
vs_EPSBCV - "Show me battery sea's voltage"
vt_EPSBCV - "Tell me battery sea's voltage"
vw_EPSBCV - "What is battery sea's voltage"
vs_EPSBCV - "Show me the voltage in battery sea"
vt_EPSBCV - "Tell me the voltage in battery sea"
vw_EPSBCV - "What is the voltage in battery sea"

EPSDBV

vw_EPSDBV - "ee pea ess dee bee vee"
vw_EPSDBV - "shunt drive voltage"
vw_EPSDBV - "voltage at the shunt drive"
vs_EPSDBV - "Show me ee pea ess dee bee vee"
vt_EPSDBV - "Tell me ee pea ess dee bee vee"
vw_EPSDBV - "What is ee pea ess dee bee vee"
vs_EPSDBV - "Show me the value of ee pea ess dee bee vee"
vt_EPSDBV - "Tell me the value of ee pea ess dee bee vee"
vw_EPSDBV - "What is the value of ee pea ess dee bee vee"
vs_EPSDBV - "Show me the shunt drive voltage"
vt_EPSDBV - "Tell me the shunt drive voltage"
vw_EPSDBV - "What is the shunt drive voltage"
vs_EPSDBV - "Show me the voltage at the shunt drive"
vt_EPSDBV - "Tell me the voltage at the shunt drive"

vw_EPSDBV - "What is the voltage at the shunt drive"

EPSEBT
vw_EPSEBT - "ee pea ess ee bee tea"
vw_EPSEBT - "temperature at shunt element bee"
vw_EPSEBT - "shunt element bee's temperature"
vs_EPSEBT - "Show me ee pea ess ee bee tea"
vt_EPSEBT - "Tell me ee pea ess ee bee tea"
vw_EPSEBT - "What is ee pea ess ee bee tea"
vs_EPSEBT - "Show me the value of ee pea ess ee bee tea"
vt_EPSEBT - "Tell me the value of ee pea ess ee bee tea"
vw_EPSEBT - "What is the value of ee pea ess ee bee tea"
vs_EPSEBT - "Show me the temperature at shunt element bee"
vt_EPSEBT - "Tell me the temperature at shunt element bee"
vw_EPSEBT - "What is the temperature at shunt element bee"
vs_EPSEBT - "Show me shunt element bee's temperature"
vt_EPSEBT - "Tell me shunt element bee's temperature"
vw_EPSEBT - "What is shunt element bee's temperature"

EPSEDT
vw_EPSEDT - "ee pea ess ee dee tea"
vw_EPSEDT - "temperature at shunt element dee"
vw_EPSEDT - "shunt element dee's temperature"
vs_EPSEDT - "Show me ee pea ess ee dee tea"
vt_EPSEDT - "Tell me ee pea ess ee dee tea"
vw_EPSEDT - "What is ee pea ess ee dee tea"
vs_EPSEDT - "Show me the value of ee pea ess ee dee tea"
vt_EPSEDT - "Tell me the value of ee pea ess ee dee tea"
vw_EPSEDT - "What is the value of ee pea ess ee dee tea"
vs_EPSEDT - "Show me the temperature at shunt element dee"
vt_EPSEDT - "Tell me the temperature at shunt element dee"
vw_EPSEDT - "What is the temperature at shunt element dee"
vs_EPSEDT - "Show me shunt element dee's temperature"
vt_EPSEDT - "Tell me shunt element dee's temperature"

vw_EPSEDT - "What is shunt element dee's temperature"

EPSEET
vw_EPSEET - "ee pea ess ee ee tea"
vw_EPSEET - "temperature at shunt element ee"
vw_EPSEET - "shunt element ee's temperature"
vs_EPSEET - "Show me ee pea ess ee ee tea"
vt_EPSEET - "Tell me ee pea ess ee ee tea"
vw_EPSEET - "What is ee pea ess ee ee tea"
vs_EPSEET - "Show me the value of ee pea ess ee ee tea"
vt_EPSEET - "Tell me the value of ee pea ess ee ee tea"
vw_EPSEET - "What is the value of ee pea ess ee ee tea"
vs_EPSEET - "Show me the temperature at shunt element ee"
vt_EPSEET - "Tell me the temperature at shunt element ee"
vw_EPSEET - "What is the temperature at shunt element ee"
vs_EPSEET - "Show me shunt element ee's temperature"
vt_EPSEET - "Tell me shunt element ee's temperature"
vw_EPSEET - "What is shunt element ee's temperature"

EPSLBI
vw_EPSLBI - "ee pea ess ell bee eye"
vw_EPSLBI - "sensor load bus current"
vw_EPSLBI - "current at the sensor load bus"
vs_EPSLBI - "Show me ee pea ess ell bee eye"
vt_EPSLBI - "Tell me ee pea ess ell bee eye"
vw_EPSLBI - "What is ee pea ess ell bee eye"
vs_EPSLBI - "Show me the value of ee pea ess ell bee eye"
vt_EPSLBI - "Tell me the value of ee pea ess ell bee eye"
vw_EPSLBI - "What is the value of ee pea ess ell bee eye"
vs_EPSLBI - "Show me the sensor load bus current"
vt_EPSLBI - "Tell me the sensor load bus current"
vw_EPSLBI - "What is the sensor load bus current"
vs_EPSLBI - "Show me the current at the sensor load bus"
vt_EPSLBI - "Tell me the current at the sensor load bus"

vw_EPSLBI - "What is the current at the sensor load bus"

EPSPBV
vw_EPSPBV - "ee pea ess pea bee vee"
vw_EPSPBV - "primary bus voltage"
vw_EPSPBV - "voltage at the primary bus"
vs_EPSPBV - "Show me ee pea ess pea bee vee"
vt_EPSPBV - "Tell me ee pea ess pea bee vee"
vw_EPSPBV - "What is ee pea ess pea bee vee"
vs_EPSPBV - "Show me the value of ee pea ess pea bee vee"
vt_EPSPBV - "Tell me the value of ee pea ess pea bee vee"
vw_EPSPBV - "What is the value of ee pea ess pea bee vee"
vs_EPSPBV - "Show me the primary bus voltage"
vt_EPSPBV - "Tell me the primary bus voltage"
vw_EPSPBV - "What is the primary bus voltage"
vs_EPSPBV - "Show me the voltage at the primary bus"
vt_EPSPBV - "Tell me the voltage at the primary bus"
vw_EPSPBV - "What is the voltage at the primary bus"

EPUC1T
vw_EPUC1T - "ee pea you sea one tea"
vt_EPUC1T - "solar array number one upper conic temperature"
vw_EPUC1T - "upper conic temperature of solar array number one"
vs_EPUC1T - "Show me ee pea you sea one tea"
vt_EPUC1T - "Tell me ee pea you sea one tea"
vw_EPUC1T - "What is ee pea you sea one tea"
vs_EPUC1T - "Show me the value of ee pea you sea one tea"
vt_EPUC1T - "Tell me the value of ee pea you sea one tea"
vw_EPUC1T - "What is the value of ee pea you sea one tea"
vs_EPUC1T - "Show me the solar array number one upper conic temperature"
vs_EPUC1T - "Tell me the solar array number one upper conic temperature"
vt_EPUC1T - "What is the solar array number one upper conic temperature"
vs_EPUC1T - "Show me the upper conic temperature of solar array number one"
vt_EPUC1T - "Tell me the upper conic temperature of solar array number one"

vw_EPUC1T - "What is the upper conic temperature of solar array number one"

<u>EPUC2T</u>

vw_EPUC2T - "ee pea you sea two tea"
vw_EPUC2T - "solar array number two upper conic temperature"
vw_EPUC2T - "upper conic temperature of solar array number two"
vs_EPUC2T - "Show me ee pea you sea two tea"
vt_EPUC2T - "Tell me ee pea you sea two tea"
vw_EPUC2T - "What is ee pea you sea two tea"
vs_EPUC2T - "Show me the value of ee pea you sea two tea"
vt_EPUC2T - "Tell me the value of ee pea you sea two tea"
vw_EPUC2T - "What is the value of ee pea you sea two tea"
vs_EPUC2T - "Show me the solar array number two upper conic temperature"
vt_EPUC2T - "Tell me the solar array number two upper conic temperature"
vw_EPUC2T - "What is the solar array number two upper conic temperature"
vs_EPUC2T - "Show me the upper conic temperature of solar array number two "
vt_EPUC2T - "Tell me the upper conic temperature of solar array number two "
vw_EPUC2T - "What is the upper conic temperature of solar array number two"

<u>EPUN1T</u>

vw_EPUN1T - "ee pea you en one tea"
vw_EPUN1T - "solar array upper narrow cylinder temperature"
vw_EPUN1T - "temperature of the solar array upper narrow cylinder"
vs_EPUN1T - "Show me ee pea you en one tea"
vt_EPUN1T - "Tell me ee pea you en one tea"
vw_EPUN1T - "What is ee pea you en one tea"
vs_EPUN1T - "Show me the value of ee pea you en one tea"
vt_EPUN1T - "Tell me the value of ee pea you en one tea"
vw_EPUN1T - "What is the value of ee pea you en one tea"
vs_EPUN1T - "Show me the solar array upper narrow cylinder temperature"
vt_EPUN1T - "Tell me the solar array upper narrow cylinder temperature"
vw_EPUN1T - "What is the solar array upper narrow cylinder temperature"
vs_EPUN1T - "Show me the temperature of the solar array upper narrow cylinder"
vt_EPUN1T - "Tell me the temperature of the solar array upper narrow cylinder"

vw_EPUN1T - "What is the temperature of the solar array upper narrow cylinder"

ESCLBI

vw_ESCLBI - "ee ess sea ell bee eye"
vw_ESCLBI - "spacecraft load bus current"
vw_ESCLBI - "current on the spacecraft load bus"
vs_ESCLBI - "Show me ee ess sea ell bee eye"
vt_ESCLBI - "Tell me ee ess sea ell bee eye"
vw_ESCLBI - "What is ee ess sea ell bee eye"
vs_ESCLBI - "Show me the value of ee ess sea ell bee eye"
vt_ESCLBI - "Tell me the value of ee ess sea ell bee eye"
vw_ESCLBI - "What is the value of ee ess sea ell bee eye"
vs_ESCLBI - "Show me the spacecraft load bus current"
vt_ESCLBI - "Tell me the spacecraft load bus current"
vw_ESCLBI - "What is the spacecraft load bus current"
vs_ESCLBI - "Show me the current on the spacecraft load bus"
vt_ESCLBI - "Tell me the current on the spacecraft load bus"
vw_ESCLBI - "What is the current on the spacecraft load bus"

ESP31T

vw_ESP31T - "ee ess pea three one tea"
vw_ESP31T - "first solar array temperature"
vw_ESP31T - "temperature of the first solar array"
vs_ESP31T - "Show me ee ess pea three one tea"
vt_ESP31T - "Tell me ee ess pea three one tea"
vw_ESP31T - "What is ee ess pea three one tea"
vs_ESP31T - "Show me the value of ee ess pea three one tea"
vt_ESP31T - "Tell me the value of ee ess pea three one tea"
vw_ESP31T - "What is the value of ee ess pea three one tea"
vs_ESP31T - "Show me the first solar array temperature"
vt_ESP31T - "Tell me the first solar array temperature"
vw_ESP31T - "What is the first solar array temperature"
vs_ESP31T - "Show me the temperature of the first solar array"
vt_ESP31T - "Tell me the temperature of the first solar array"

vw_ESP31T - "What is the temperature of the first solar array"

<u>ESP32T</u>

vw_ESP32T - "ee ess pea three two tea"
vw_ESP32T - "second solar array temperature"
vw_ESP32T - "temperature of the second solar array"
vs_ESP32T - "Show me ee ess pea three two tea"
vt_ESP32T - "Tell me ee ess pea three two tea"
vw_ESP32T - "What is ee ess pea three two tea"
vs_ESP32T - "Show me the value of ee ess pea three two tea"
vt_ESP32T - "Tell me the value of ee ess pea three two tea"
vw_ESP32T - "What is the value of ee ess pea three two tea"
vs_ESP32T - "Show me the second solar array temperature"
vt_ESP32T - "Tell me the second solar array temperature"
vw_ESP32T - "What is the second solar array temperature"
vs_ESP32T - "Show me the temperature of the second solar array"
vt_ESP32T - "Tell me the temperature of the second solar array"
vw_ESP32T - "What is the temperature of the second solar array"

<u>SLOART</u>

vw_SLOART - "ess ell oh eh are tea"
vw_SLOART - "slow art"
vw_SLOART - "temperature of the lower solar array structure"
vw_SLOART - "lower solar array structure's temperature"
vs_SLOART - "Show me ess ell oh eh are tea"
vt_SLOART - "Tell me ess ell oh eh are tea"
vw_SLOART - "What is ess ell oh eh are tea"
vs_SLOART - "Show me the value of ess ell oh eh are tea"
vt_SLOART - "Tell me the value of ess ell oh eh are tea"
vw_SLOART - "What is the value of ess ell oh eh are tea"
vs_SLOART - "Show me slow art"
vt_SLOART - "Tell me slow art"
vw_SLOART - "What is slow art"
vs_SLOART - "Show me the value of slow art"

vt_SLOART - "Tell me the value of slow art"
vw_SLOART - "What is the value of slow art"
vs_SLOART - "Show me the temperature of the lower solar array structure"
vt_SLOART - "Tell me the temperature of the lower solar array structure"
vw_SLOART - "What is the temperature of the lower solar array structure"
vs_SLOART - "Show me the lower solar array structure's temperature"
vt_SLOART - "Tell me the lower solar array structure's temperature"
vw_SLOART - "What is the lower solar array structure's temperature"

SPXCPT
vw_SPXCPT - "ess pea ex sea pea tea"
vw_SPXCPT - "temperature of the structural platform"
vw_SPXCPT - "structural platform's temperature"
vs_SPXCPT - "Show me ess pea ex sea pea tea"
vt_SPXCPT - "Tell me ess pea ex sea pea tea"
vw_SPXCPT - "What is ess pea ex sea pea tea"
vs_SPXCPT - "Show me the value of ess pea ex sea pea tea"
vt_SPXCPT - "Tell me the value of ess pea ex sea pea tea"
vw_SPXCPT - "What is the value of ess pea ex sea pea tea"
vs_SPXCPT - "Show me the temperature of the structural platform"
vt_SPXCPT - "Tell me the temperature of the structural platform"
vw_SPXCPT - "What is the temperature of the structural platform"
vs_SPXCPT - "Show me the structural platform's temperature"
vt_SPXCPT - "Tell me the structural platform's temperature"
vw_SPXCPT - "What is the structural platform's temperature"

DISCRETE COMMANDS

LINK 2 COMMUNICATIONS SUBSYSTEM

C2ASPB
C2ASPB_1 - "change sea two eh ess pea bee from eh high to bee high"
C2ASPB_0 - "change sea two eh ess pea bee from bee high to eh high"
C2ASPB_1 - "change the antenna switch position monitor from eh high to bee high"
C2ASPB_0 - "change the antenna switch position monitor from bee high to eh high"
C2ASPB_0 - "set sea two eh ess pea bee to eh high"
C2ASPB_1 - "set sea two eh ess pea bee to bee high"
C2ASPB_0 - "set the antenna switch position monitor to eh high"
C2ASPB_1 - "set the antenna switch position monitor to bee high"

EDABRB
EDABRB_1 - "change ee dee eh bee are bee from one kay to one hundred twenty eight kay"
EDABRB_1 - "change ee dee eh bee are bee from one kay to one twenty eight kay"
EDABRB_1 - "change ee dee eh bee are bee from one kay to one two eight kay"
EDABRB_0 - "change ee dee eh bee are bee from one hundred twenty eight kay to one kay"
EDABRB_0 - "change ee dee eh bee are bee from one twenty eight kay to one kay"
EDABRB_0 - "change ee dee eh bee are bee from one two eight kay to one kay"
EDABRB_1 - "change the digital telemetry unit eh bit rate from one kay to one hundred twenty eight kay"
EDABRB_1 - "change the digital telemetry unit eh bit rate from one kay to one twenty eight kay"
EDABRB_1 - "change the digital telemetry unit eh bit rate from one kay to one two eight kay"
EDABRB_0 - "change the digital telemetry unit eh bit rate from one hundred twenty eight kay to one kay"
EDABRB_0 - "change the digital telemetry unit eh bit rate from one twenty eight kay to one kay"
EDABRB_0 - "change the digital telemetry unit eh bit rate from one two eight kay to one kay"
EDABRB_1 - "set ee dee eh bee are bee to one hundred twenty eight kay"
EDABRB_1 - "set ee dee eh bee are bee to one twenty eight kay"
EDABRB_1 - "set ee dee eh bee are bee to one two eight kay"
EDABRB_0 - "set ee dee eh bee are bee to one kay"

EDABRB_1 - "set the digital telemetry unit eh bit rate to one hundred twenty eight kay"
EDABRB_1 - "set the digital telemetry unit eh bit rate to one twenty eight kay"
EDABRB_1 - "set the digital telemetry unit eh bit rate to one two eight kay"
EDABRB_0 - "set the digital telemetry unit eh bit rate to one kay"

<u>EDBBRB</u>
EDBBRB_1 - "change ee dee bee bee are bee from one kay to one hundred twenty eight kay"
EDBBRB_1 - "change ee dee bee bee are bee from one kay to one twenty eight kay"
EDBBRB_1 - "change ee dee bee bee are bee from one kay to one two eight kay"
EDBBRB_0 - "change ee dee bee bee are bee from one hundred twenty eight kay to one kay"
EDBBRB_0 - "change ee dee bee bee are bee from one twenty eight kay to one kay"
EDBBRB_0 - "change ee dee bee bee are bee from one two eight kay to one kay"
EDBBRB_1 - "change the digital telemetry unit bee bit rate from one kay to one hundred twenty eight kay"
EDBBRB_1 - "change the digital telemetry unit bee bit rate from one kay to one twenty eight kay"
EDBBRB_1 - "change the digital telemetry unit bee bit rate from one kay to one two eight kay"
EDBBRB_0 - "change the digital telemetry unit bee bit rate from one hundred twenty eight kay to one kay"
EDBBRB_0 - "change the digital telemetry unit bee bit rate from one twenty eight kay to one kay"
EDBBRB_0 - "change the digital telemetry unit bee bit rate from one two eight kay to one kay"
EDBBRB_1 - "set ee dee bee bee are bee to one hundred twenty eight kay"
EDBBRB_1 - "set ee dee bee bee are bee to one twenty eight kay"
EDBBRB_1 - "set ee dee bee bee are bee to one two eight kay"
EDBBRB_0 - "set ee dee bee bee are bee to one kay"
EDBBRB_1 - "set digital telemetry unit bee bit rate to one hundred twenty eight kay"
EDBBRB_1 - "set digital telemetry unit bee bit rate to one twenty eight kay"
EDBBRB_1 - "set digital telemetry unit bee bit rate to one two eight kay"
EDBBRB_0 - "set digital telemetry unit bee bit rate to one kay"

<u>EDTUAB</u>
EDTUAB_1 - "change ee dee tea you eh bee from off to on"
EDTUAB_0 - "change ee dee tea you eh bee from on to off"
EDTUAB_1 - "change digital telemetry unit eh from off to on"
EDTUAB_0 - "change digital telemetry unit eh from on to off"
EDTUAB_1 - "set ee dee tea you eh bee to on"
EDTUAB_0 - "set ee dee tea you eh bee to off"

EDTUAB_1 - "set digital telemetry unit eh to on"
EDTUAB_0 - "set digital telemetry unit eh to off"
EDTUAB_1 - "turn on ee dee tea you eh bee"
EDTUAB_0 - "turn off ee dee tea you eh bee"
EDTUAB_1 - "turn on digital telemetry unit eh"
EDTUAB_0 - "turn off digital telemetry unit eh"
EDTUAB_1 - "turn ee dee tea you eh bee on"
EDTUAB_0 - "turn ee dee tea you eh bee off"
EDTUAB_1 - "turn digital telemetry unit eh on"
EDTUAB_0 - "turn digital telemetry unit eh off"

EDTUBB
EDTUBB_1 - "change ee dee tea you bee bee from off to on"
EDTUBB_0 - "change ee dee tea you bee bee from on to off"
EDTUBB_1 - "change digital telemetry unit bee from off to on"
EDTUBB_0 - "change digital telemetry unit bee from on to off"
EDTUBB_1 - "set ee dee tea you bee bee to on"
EDTUBB_0 - "set ee dee tea you bee bee to off"
EDTUBB_1 - "set digital telemetry unit bee to on"
EDTUBB_0 - "set digital telemetry unit bee to off"
EDTUBB_1 - "turn on ee dee tea you bee bee"
EDTUBB_0 - "turn off ee dee tea you bee bee"
EDTUBB_1 - "turn on digital telemetry unit bee"
EDTUBB_0 - "turn off digital telemetry unit bee"
EDTUBB_1 - "turn ee dee tea you bee bee on"
EDTUBB_0 - "turn ee dee tea you bee bee off"
EDTUBB_1 - "turn digital telemetry unit bee on"
EDTUBB_0 - "turn digital telemetry unit bee off"

EDTUDB
EDTUDB_1 - "change ee dee tea you dee bee from bye pass to encrypt"
EDTUDB_0 - "change ee dee tea you dee bee from encrypt to bye pass"
EDTUDB_1 - "change digital telemetry unit eh bee from bye pass to encrypt"
EDTUDB_0 - "change digital telemetry unit eh bee from encrypt to bye pass"

EDTUDB_1 - "set ee dee tea you dee bee to encrypt"
EDTUDB_0 - "set ee dee tea you dee bee to bye pass"
EDTUDB_1 - "set digital telemetry unit eh bee to encrypt"
EDTUDB_0 - "set digital telemetry unit eh bee to bye pass"

EKG2AB
EKG2AB_1 - "change ee kay gee two eh bee from off to on"
EKG2AB_0 - "change ee kay gee two eh bee from on to off"
EKG2AB_1 - "change ee kay gee ex twenty eight eh from off to on"
EKG2AB_1 - "change kay gee ex two eight eh from off to on"
EKG2AB_0 - "change kay gee ex twenty eight eh from on to off"
EKG2AB_0 - "change kay gee ex two eight eh from on to off"
EKG2AB_1 - "set ee kay gee two eh bee to on"
EKG2AB_0 - "set ee kay gee two eh bee to off"
EKG2AB_1 - "set kay gee ex twenty eight eh to on"
EKG2AB_1 - "set kay gee ex two eight eh to on"
EKG2AB_0 - "set kay gee ex twenty eight eh to off"
EKG2AB_0 - "set kay gee ex two eight eh to off"
EKG2AB_1 - "turn on ee kay gee two eh bee"
EKG2AB_0 - "turn off ee kay gee two eh bee"
EKG2AB_1 - "turn on kay gee ex twenty eight"
EKG2AB_0 - "turn off kay gee ex two eight eh"
EKG2AB_0 - "turn off kay gee ex twenty eight eh"
EKG2AB_1 - "turn on kay gee ex two eight eh"
EKG2AB_1 - "turn ee kay gee two eh bee on"
EKG2AB_0 - "turn ee kay gee two eh bee off"
EKG2AB_0 - "turn kay gee ex twenty eight off"
EKG2AB_0 - "turn kay gee ex two eight eh off"
EKG2AB_1 - "turn kay gee ex twenty eight on"
EKG2AB_1 - "turn kay gee ex two eight on on"

EKG2BB
EKG2BB_1 - "change ee kay gee two bee bee from off to on"
EKG2BB_0 - "change ee kay gee two bee bee from on to off"

EKG2BB_1 - "change kay gee ex twenty eight bee from off to on"
EKG2BB_1 - "change kay gee ex two eight bee from off to on"
EKG2BB_0 - "change kay gee ex twenty eight bee from on to off"
EKG2BB_0 - "change kay gee ex two eight bee from on to off"
EKG2BB_1 - "set ee kay gee two bee bee to on"
EKG2BB_0 - "set ee kay gee two bee bee to off"
EKG2BB_1 - "set kay gee ex twenty eight bee to on"
EKG2BB_1 - "set kay gee ex two eight bee to on"
EKG2BB_0 - "set kay gee ex twenty eight bee to off"
EKG2BB_0 - "set kay gee ex two eight bee to off"
EKG2BB_1 - "turn on ee kay gee two bee bee"
EKG2BB_0 - "turn off ee kay gee two bee bee"
EKG2BB_1 - "turn on kay gee ex twenty eight bee"
EKG2BB_1 - "turn on kay gee ex two eight bee"
EKG2BB_0 - "turn off kay gee ex twenty eight bee"
EKG2BB_0 - "turn off kay gee ex two eight bee"
EKG2BB_1 - "turn ee kay gee two bee bee on"
EKG2BB_0 - "turn ee kay gee two bee bee off"
EKG2BB_1 - "turn kay gee ex twenty eight bee on"
EKG2BB_1 - "turn kay gee ex two eight bee on"
EKG2BB_0 - "turn kay gee ex twenty eight bee off"
EKG2BB_0 - "turn kay gee ex two eight bee off"

<u>ERBUDB</u>
ERBUDB_1 - "change ee are bee you dee bee from bye pass to encrypt"
ERBUDB_0 - "change ee are bee you dee bee from encrypt to bye pass"
ERBUDB_1 - "change the digital redundant baseband assembly unit status from bye pass to encrypt"
ERBUDB_0 - "change the digital redundant baseband assembly unit status from encrypt to bye pass"
ERBUDB_1 - "set ee are bee you dee bee to encrypt"
ERBUDB_0 - "set ee are bee you dee bee to bye pass"
ERBUDB_1 - "set the digital redundant baseband assembly unit status to encrypt"
ERBUDB_0 - "set the digital redundant baseband assembly unit status to bye pass"

<u>ET2AAB</u>
ET2AAB_1 - "change ee tea two eh eh bee from safe to arm"
ET2AAB_0 - "chang e ee tea two eh eh bee from arm to safe"
ET2AAB_1 - "change transmitter eh from safe to arm"
ET2AAB_0 - "change transmitter eh from arm to safe"
ET2AAB_1 - "change ee tea two eh eh bee to arm"
ET2AAB_1 - "change ee tea two eh eh bee to arm"
ET2AAB_1 - "set ee tea two eh eh bee to arm"
ET2AAB_0 - "set ee tea two eh eh bee to safe"
ET2AAB_1 - "set transmitter eh to arm"
ET2AAB_0 - "set transmitter eh to safe"
ET2AAB_1 - "arm ee tea two eh eh bee"
ET2AAB_0 - "safe ee tea two eh eh bee"

<u>ET2AEB</u>
ET2AEB_1 - "change ee tea two eh ee bee from disabled to enabled"
ET2AEB_0 - "change ee tea two eh ee bee from enabled to disabled"
ET2AEB_1 - "change transmitter eh from disabled to enabled"
ET2AEB_0 - "change transmitter eh from enabled to disabled"
ET2AEB_1 - "set ee tea two eh ee bee to enabled"
ET2AEB_0 - "set ee tea two eh ee bee to disabled"
ET2AEB_1 - "set transmitter eh to enabled"
ET2AEB_0 - "set transmitter eh to disabled"
ET2AEB_0 - "disable ee tea two eh ee bee"), "","", varEmpty, varEmpty);
ET2AEB_1 - "enable ee tea two eh ee bee"), "","", varEmpty, varEmpty);
ET2AEB_0 - "disable transmitter eh"), "","", varEmpty, varEmpty);
ET2AEB_1 - "enable transmitter eh"), "","", varEmpty, varEmpty);

<u>ET2AOB</u>
ET2AOB_1 - "change ee tea two eh oh bee from off to on"
ET2AOB_0 - "change ee tea two eh oh bee from on to off"
ET2AOB_1 - "change transmitter eh from off to on"
ET2AOB_0 - "change transmitter eh from on to off"
ET2AOB_1 - "set ee tea two eh oh bee to on"

ET2AOB_0 - "set ee tea two eh oh bee to off"
ET2AOB_1 - "set transmitter eh to on"
ET2AOB_0 - "set transmitter eh to off"
ET2AOB_1 - "turn ee tea two eh oh bee on"
ET2AOB_0 - "turn ee tea two eh oh bee off"
ET2AOB_1 - "turn transmitter eh on"
ET2AOB_0 - "turn transmitter eh off"
ET2AOB_1 - "turn on ee tea two eh oh bee"
ET2AOB_0 - "turn off ee tea two eh oh bee"
ET2AOB_1 - "turn on transmitter eh"
ET2AOB_0 - "turn off transmitter eh"

<u>ET2BAB</u>
ET2BAB_1 - "change ee tea two bee eh bee from safe to arm"
ET2BAB_0 - "change ee tea two bee eh bee from arm to safe"
ET2BAB_1 - "change transmitter bee from safe to arm"
ET2BAB_0 - "change transmitter bee from arm to safe"
ET2BAB_1 - "set ee tea two bee eh bee to arm"
ET2BAB_0 - "set ee tea two bee eh bee to safe"
ET2BAB_1 - "set transmitter bee to arm"
ET2BAB_0 - "set transmitter bee to safe"

<u>ET2BEB</u>
ET2BEB_1 - "change ee tea two bee ee bee from disabled to enabled"
ET2BEB_0 - "change ee tea two bee ee bee from enabled to disabled"
ET2BEB_1 - "change transmitter bee from disabled to enabled"
ET2BEB_0 - "change transmitter be from enabled to disabled"
ET2BEB_1 - "set ee tea two bee ee bee to enabled"
ET2BEB_0 - "set ee tea two bee ee bee to disabled"
ET2BEB_1 - "set transmitter bee to enabled"
ET2BEB_0 - "set transmitter bee to disabled"
ET2BEB_1 - "enable ee tea two bee ee bee"
ET2BEB_0 - "disable ee tea two bee ee bee"
ET2BEB_1 - "enable transmitter bee"

ET2BEB_0 - "disable transmitter bee"

ET2BOB

ET2BOB_1 - "change ee tea two bee oh bee from off to on"
ET2BOB_0 - "change ee tea two bee oh bee from off to off"
ET2BOB_1 - "change transmitter bee from off to on"
ET2BOB_0 - "change transmitter bee from on to off"
ET2BOB_1 - "set ee tea two bee oh bee to on"
ET2BOB_0 - "set ee tea two bee oh bee to off"
ET2BOB_1 - "set transmitter bee to on"
ET2BOB_0 - "set transmitter bee to off"
ET2BOB_1 - "turn ee tea two bee oh bee on"
ET2BOB_0 - "turn ee tea two bee oh bee off"
ET2BOB_1 - "turn transmitter bee on"
ET2BOB_0 - "turn transmitter bee off"
ET2BOB_1 - "turn on ee tea two bee oh bee"
ET2BOB_0 - "turn off ee tea two bee oh bee"
ET2BOB_1 - "turn on transmitter bee"
ET2BOB_0 - "turn off transmitter bee"

LINK 1 COMMUNICATIONS SUBSYSTEM

CDCAMB

CDCAMB_1 - "change sea dee sea eh em bee from bye pass to code"
CDCAMB_0 - "change sea dee sea eh em bee from code to bye pass"
CDCAMB_1 - "change dual error coder eh from bye pass to code"
CDCAMB_0 - "change dual error coder eh from code to bye pass"
CDCAMB_1 - "set sea dee sea eh em bee to code"
CDCAMB_0 - "set sea dee sea eh em bee to bye pass"
CDCAMB_1 - "set dual error coder eh to code"
CDCAMB_0 - "set dual error coder eh to bye pass"

CDCBMB
CDCBMB_1 - "change sea dee sea bee em bee from bye pass to code"
CDCBMB_0 - "change sea dee sea bee em bee from code to bye pass"
CDCBMB_1 - "change dual error coder bee from bye pass to code"
CDCBMB_0 - "change dual error coder bee from code to bye pass"
CDCBMB_1 - "set sea dee sea bee em bee to code"
CDCBMB_0 - "set sea dee sea bee em bee to bye pass"
CDCBMB_1 - "set dual error coder bee to code"
CDCBMB_0 - "set dual error coder bee to bye pass"

CSW1PB
CSW1PB_1 - "change sea ess double you one pea bee from two point five watts to twenty watts"
CSW1PB_1 - "change sea ess double you one pea bee from two and a half watts to twenty watts"
CSW1PB_0 - "change sea ess double you one pea bee from twenty watts to two point five watts"
CSW1PB_0 - "change sea ess double you one pea bee from twenty watts to two and a half watts"
CSW1PB_1 - "change the transmitter eh switch one position from two point five watts to twenty watts"
CSW1PB_1 - "change the transmitter eh switch one position from two and a half watts to twenty watts"
CSW1PB_0 - "change the transmitter eh switch one position from twenty watts to two point five watts"
CSW1PB_0 - "change the transmitter eh switch one position from twenty watts to two and a half watts"
CSW1PB_1 - "set sea ess double you one pea bee to twenty watts"
CSW1PB_0 - "set sea ess double you one pea bee to two point five watts"
CSW1PB_0 - "set sea ess double you one pea bee to two and a half watts"
CSW1PB_1 - "set transmitter eh switch one to twenty watts"
CSW1PB_0 - "set transmitter eh switch one to two point five watts"
CSW1PB_0 - "set transmitter eh switch one to two and a half five watts"

CSW2PB
CSW2PB_1 - "change sea ess double you two pea bee from two point five watts to twenty watts"
CSW2PB_1 - "change sea ess double you two pea bee from two and a half watts to twenty watts"
CSW2PB_0 - "change sea ess double you two pea bee from twenty watts to two point five watts"
CSW2PB_0 - "change sea ess double you two pea bee from twenty watts to two and a half watts"
CSW2PB_1 - "change the transmitter eh switch two position from two point five watts to twenty watts"
CSW2PB_1 - "change the transmitter eh switch two position from two and a half watts to twenty watts"
CSW2PB_0 - "change the transmitter eh switch two position from twenty watts to two point five watts"

CSW2PB_0 - "change the transmitter eh switch two position from twenty watts to two and a half watts"
CSW2PB_1 - "set sea ess double you two pea bee to twenty watts"
CSW2PB_0 - "set sea ess double you two pea bee to two point five watts"
CSW2PB_0 - "set sea ess double you two pea bee to two and a half watts"
CSW2PB_1 - "set transmitter eh switch two to twenty watts"
CSW2PB_0 - "set transmitter eh switch two to two point five watts"
CSW2PB_0 - "set transmitter eh switch two to two and a half watts"

<u>CSW3PB</u>
CSW3PB_0 - "change sea ess double you three pea bee from eh to bee"
CSW3PB_1 - "change sea ess double you three pea bee from bee to eh"
CSW3PB_0 - "change the link one channel from eh to bee"
CSW3PB_1 - "change the link one channel from bee to eh"
CSW3PB_0 - "set sea ess double you three pea bee to bee"
CSW3PB_1 - "set sea ess double you three pea bee to eh"
CSW3PB_0 - "set the link one channel to bee"
CSW3PB_1 - "set the link one channel to eh"

<u>CSW4PB</u>
CSW4PB_1 - "change sea ess double you four pea bee from two point five watts to twenty watts"
CSW4PB_1 - "change sea ess double you four pea bee from two and a half watts to twenty watts"
CSW4PB_0 - "change sea ess double you four pea bee from twenty watts to two point five watts"
CSW4PB_0 - "change sea ess double you four pea bee from twenty watts to two and a half watts"
CSW4PB_1 - "change the transmitter bee switch four position from two point five watts to twenty watts"
CSW4PB_1 - "change the transmitter bee switch four position from two and a half watts to twenty watts"
CSW4PB_0 - "change the transmitter bee switch four position from twenty watts to two point five watts"
CSW4PB_0 - "change the transmitter bee switch four position from twenty watts to two and a half watts"
CSW4PB_1 - "set sea ess double you four pea bee to twenty watts"
CSW4PB_0 - "set sea ess double you four pea bee to two point five watts"
CSW4PB_0 - "set sea ess double you four pea bee to two and a half watts"
CSW4PB_1 - "set transmitter bee switch four to twenty watts"
CSW4PB_0 - "set transmitter bee switch four to two point five watts"
CSW4PB_0 - "set transmitter bee switch four to two and a half watts"

<u>CSW5PB</u>
CSW5PB_1 - "change sea ess double you five pea bee from two point five watts to twenty watts"
CSW5PB_1 - "change sea ess double you five pea bee from two and a half watts to twenty watts"
CSW5PB_0 - "change sea ess double you five pea bee from twenty watts to two point five watts"
CSW5PB_0 - "change sea ess double you five pea bee from twenty watts to two and a half watts"
CSW5PB_1 - "change the transmitter bee switch five position from two point five watts to twenty watts"
CSW5PB_1 - "change the transmitter bee switch five position from two and a half watts to twenty watts"
CSW5PB_0 - "change the transmitter bee switch five position from twenty watts to two point five watts"
CSW5PB_0 - "change the transmitter bee switch five position from twenty watts to two and a half watts"
CSW5PB_1 - "set sea ess double you five pea bee to twenty watts"
CSW5PB_0 - "set sea ess double you five pea bee to two point five watts"
CSW5PB_0 - "set sea ess double you five pea bee to two and a half watts"
CSW5PB_1 - "set transmitter bee switch five to twenty watts"
CSW5PB_0 - "set transmitter bee switch five to two point five watts"
CSW5PB_0 - "set transmitter bee switch five to two and a half watts"

<u>E1ASOB</u>
E1ASOB_1 - "change ee one eh ess oh bee from overide to normal"
E1ASOB_0 - "change ee one eh ess oh bee from normal to overide"
E1ASOB_1 - "change the link one eh inhibit switch from overide to normal"
E1ASOB_0 - "change the link one eh inhibit switch from normal to overide"
E1ASOB_1 - "set ee one eh ess oh bee to normal"
E1ASOB_0 - "set ee one eh ess oh bee to overide"
E1ASOB_1 - "set the link one eh inhibit switch to normal"
E1ASOB_0 - "set the link one eh inhibit switch to overide"

<u>E1BSOB</u>
E1BSOB_1 - "change ee one bee ess oh bee from overide to normal"
E1BSOB_0 - "change ee one eh bee ess oh bee from normal to overide"
E1BSOB_1 - "change the link one bee inhibit switch from overide to normal"
E1BSOB_0 - "change the link one bee inhibit switch from normal to overide"
E1BSOB_1 - "set ee one bee ess oh bee to normal"
E1BSOB_0 - "set ee one bee ess oh bee to overide"
E1BSOB_1 - "set the link one bee inhibit switch to normal"

E1BSOB_0 - "set the link one bee inhibit switch to overide"

EDC1AB
EDC1AB_1 - "change ee dee sea one eh bee from encrypt to bye pass"
EDC1AB_0 - "change ee dee sea one eh bee from bye pass to encrypt"
EDC1AB_1 - "change the link one eh data control from encrypt to bye pass"
EDC1AB_0 - "change the link one eh data control from bye pass to encrypt"
EDC1AB_1 - "set ee dee sea one eh bee to bye pass"
EDC1AB_0 - "set ee dee sea one eh bee to encrypt"
EDC1AB_1 - "set the link one eh data control to bye pass"
EDC1AB_0 - "set the link one eh data control to encrypt"

EDECAB
EDECAB_1 - "change ee dee ee sea eh bee from off to on"
EDECAB_0 - "change ee dee ee sea eh bee from on to off"
EDECAB_1 - "change dual error coder eh from off to on"
EDECAB_0 - "change dual error coder eh from on to off"
EDECAB_1 - "set ee dee ee sea eh bee to on"
EDECAB_0 - "set ee dee ee sea eh bee to off"
EDECAB_1 - "set dual error coder eh to on"
EDECAB_0 - "set dual error coder eh to off"
EDECAB_1 - "turn ee dee ee sea eh bee on"
EDECAB_0 - "turn ee dee ee sea eh bee off"
EDECAB_1 - "turn dual error coder eh on"
EDECAB_0 - "turn dual error coder eh off"
EDECAB_1 - "turn on ee dee ee sea eh bee"
EDECAB_0 - "turn off ee dee ee sea eh bee"
EDECAB_1 - "turn on dual error coder eh"
EDECAB_0 - "turn off dual error coder eh"

EDECBB
EDECBB_1 - "change ee dee ee sea bee bee from off to on"
EDECBB_0 - "change ee dee ee sea bee bee from on to off"
EDECBB_1 - "change dual error coder bee from off to on"

EDECBB_0 - "change dual error coder bee from on to off"
EDECBB_1 - "set ee dee ee sea bee bee to on"
EDECBB_0 - "set ee dee ee sea bee bee to off"
EDECBB_1 - "set dual error coder bee to on"
EDECBB_0 - "set dual error coder bee to off"
EDECBB_1 - "turn ee dee ee sea bee bee on"
EDECBB_0 - "turn ee dee ee sea bee bee off"
EDECBB_1 - "turn dual error coder bee on"
EDECBB_0 - "turn dual error coder bee off"
EDECBB_1 - "turn on ee dee ee sea bee bee"
EDECBB_0 - "turn off ee dee ee sea bee bee"
EDECBB_1 - "turn on dual error bee coder"
EDECBB_0 - "turn off dual error coder bee"

<u>EKG1AB</u>
EKG1AB_1 - "change ee kay gee one eh bee from off to on"
EKG1AB_0 - "change ee kay gee one eh bee from on to off"
EKG1AB_1 - "change the link one kay gee ex twenty eight eh bee from off to on"
EKG1AB_0 - "change the link one kay gee ex twenty eight eh bee from on to off"
EKG1AB_1 - "set ee kay gee one eh bee to on"
EKG1AB_0 - "set ee kay gee one eh bee to off"
EKG1AB_1 - "set the link one kay gee ex twenty eight eh bee to on"
EKG1AB_0 - "set the link one kay gee ex twenty eight eh bee to off"
EKG1AB_1 - "turn ee kay gee one eh bee on"
EKG1AB_0 - "turn ee kay gee one eh bee off"
EKG1AB_1 - "turn link one kay gee ex twenty eight eh bee on"
EKG1AB_0 - "turn link one kay gee ex twenty eight eh bee off"
EKG1AB_1 - "turn on ee kay gee one eh bee"
EKG1AB_0 - "turn off ee kay gee one eh bee"
EKG1AB_1 - "turn on link one kay gee ex twenty eight eh bee"
EKG1AB_0 - "turnoff link one kay gee ex twenty eight eh bee"

<u>EKG1BB</u>
EKG1BB_1 - "change ee kay gee one bee bee from off to on"

EKG1BB_0 - "change ee kay gee one bee bee from on to off"
EKG1BB_1 - "change the link one kay gee ex twenty eight eh bee from off to on"
EKG1BB_0 - "change the link one kay gee ex twenty eight eh be from on to off"
EKG1BB_1 - "set ee kay gee one bee bee to on"
EKG1BB_0 - "set ee kay gee one bee bee to off"
EKG1BB_1 - "set link one kay gee ex twenty eight eh bee to on"
EKG1BB_0 - "set link one kay gee ex twenty eight eh bee to off"
EKG1BB_1 - "turn ee kay gee one bee bee on"
EKG1BB_0 - "turn ee kay gee one bee bee off"
EKG1BB_1 - "turn link one kay gee ex twenty eight eh bee on"
EKG1BB_0 - "turn link one kay gee ex twenty eight eh bee off"
EKG1BB_1 - "turn on ee kay gee one bee bee"
EKG1BB_0 - "turn off ee kay gee one bee bee"
EKG1BB_1 - "turn on link one kay gee ex twenty eight eh bee"
EKG1BB_0 - "turn off link one kay gee ex twenty eight eh bee"

<u>EP1AHB</u>
EP1AHB_1 - "change ee pea one eh H bee from disabled to enabled"
EP1AHB_0 - "change ee pea one eh H bee from enabled to disabled"
EP1AHB_1 - "change the solid state power amplifier eh heater status from disabled to enabled"
EP1AHB_0 - "change the solid state power amplifier eh heater status from enabled to disabled"
EP1AHB_1 - "set ee pea one eh H bee to enabled"
EP1AHB_0 - "set ee pea one eh H bee to disabled"
EP1AHB_1 - "set the solid state power amplifier eh heater status to enabled"
EP1AHB_0 - "set the solid state power amplifier eh heater status to disabled"
EP1AHB_1 - "enable ee pea one eh H bee"
EP1AHB_0 - "disable ee pea one eh H bee"
EP1AHB_1 - "enable solid state power amplifier eh heater"
EP1AHB_0 - "disable solid state power amplifier eh heater"

<u>EP1ASB</u>
EP1ASB_1 - "change ee pea one eh ess be from off to on"
EP1ASB_0 - "change ee pea one eh ess be from on to off"
EP1ASB_1 - "change solid state amplifier eh from off to on"

EP1ASB_0 - "change solid state amplifier eh from on to off"
EP1ASB_1 - "set ee pea one eh ess be to on"
EP1ASB_0 - "set ee pea one eh ess be to off"
EP1ASB_1 - "set solid state amplifier eh to on"
EP1ASB_0 - "set solid state amplifier eh to off"
EP1ASB_1 - "turn ee pea one eh ess be on"
EP1ASB_0 - "turn ee pea one eh ess be off"
EP1ASB_1 - "turn solid state amplifier eh on"
EP1ASB_0 - "turn solid state amplifier eh off"
EP1ASB_1 - "turn on ee pea one eh ess be"
EP1ASB_0 - "turn off ee pea one eh ess be"
EP1ASB_1 - "turn on solid state amplifier eh"
EP1ASB_0 - "turn off solid state amplifier eh"

EP1BSB
EP1BSB_1 - "change ee pea one bee ess bee from off to on"
EP1BSB_0 - "change ee pea one bee ess bee from on to off"
EP1BSB_1 - "change solid state amplifier bee from off to on"
EP1BSB_0 - "change solid state amplifier bee from on to off"
EP1BSB_1 - "set ee pea one bee ess bee to on"
EP1BSB_0 - "set ee pea one bee ess bee to off"
EP1BSB_1 - "set solid state amplifier bee to on"
EP1BSB_0 - "set solid state amplifier bee to off"
EP1BSB_1 - "turn ee pea one bee ess bee on"
EP1BSB_0 - "turn ee pea one bee ess bee off"
EP1BSB_1 - "turn solid state amplifier bee on"
EP1BSB_0 - "turn solid state amplifier bee off"
EP1BSB_1 - "turn on ee pea one bee ess bee"
EP1BSB_0 - "turn off ee pea one bee ess bee"
EP1BSB_1 - "turn on solid state amplifier bee"
EP1BSB_0 - "turn off solid state amplifier bee"

EP1BHB
EP1BHB_1 - "change ee pea one bee H bee from disabled to enabled"

EP1BHB_0 - "change ee pea one bee H bee from enabled to disabled"
EP1BHB_1 - "change the solid state power amplifier bee heater status from disabled to enabled"
EP1BHB_0 - "change the solid state power amplifier bee heater status from enabled to disabled"
EP1BHB_1 - "set ee pea one bee H bee to enabled"
EP1BHB_0 - "set ee pea one bee H bee to disabled"
EP1BHB_1 - "set the solid state power amplifier bee heater status to enabled"
EP1BHB_0 - "set the solid state power amplifier bee heater status to disabled"
EP1BHB_1 - "enable ee pea one bee H bee"
EP1BHB_0 - "disable ee pea one bee H bee"
EP1BHB_1 - "enable the solid state power amplifier bee heater "
EP1BHB_0 - "disable the solid state power amplifier bee heater"

<u>ET1AAB</u>
ET1AAB_1 - "change ee tea one eh eh bee from safe to arm"
ET1AAB_0 - "change ee tea one eh eh bee from arm to safe"
ET1AAB_1 - "change two point five watt transmitter eh from safe to arm"
ET1AAB_1 - "change two and a half watt transmitter eh from safe to arm"
ET1AAB_0 - "change two point five watt transmitter eh from arm to safe"
ET1AAB_0 - "change two and a half watt transmitter eh from arm to safe"
ET1AAB_1 - "set ee tea one eh eh bee to arm"
ET1AAB_0 - "set ee tea one eh eh bee to safe"
ET1AAB_1 - "set two point five watt transmitter eh to arm"
ET1AAB_1 - "set two point and a half watt transmitter eh to arm"
ET1AAB_0 - "set two point five watt transmitter eh to safe"
ET1AAB_0 - "set two point and a half watt transmitter eh to safe"

<u>ET1AEB</u>
ET1AEB_1 - "change ee tea one eh ee bee from enabled to disabled"
ET1AEB_0 - "change ee tea one eh ee bee from disabled to enabled"
ET1AEB_1 - "change two point five watt transmitter eh from enabled to disabled"
ET1AEB_1 - "change two and a half watt transmitter eh from enabled to disabled"
ET1AEB_0 - "change two point five watt transmitter eh from disabled to enabled"
ET1AEB_0 - "change two and a half watt transmitter eh from disabled to enabled"
ET1AEB_1 - "disable ee tea one eh ee bee"

ET1AEB_0 - "enable ee tea one eh ee bee"
ET1AEB_1 - "disable two point five watt transmitter eh"
ET1AEB_1 - "disable two and a half watt transmitter eh"
ET1AEB_0 - "enable two point five watt transmitter eh"
ET1AEB_0 - "enable two and a half watt transmitter eh"

ET1AOB
ET1AOB_1 - "change ee tea one eh oh bee from off to on"
ET1AOB_0 - "change ee tea one eh oh bee from on to off"
ET1AOB_1 - "change two point five watt transmitter eh from off to on"
ET1AOB_1 - "change two and a half watt transmitter eh from off to on"
ET1AOB_0 - "change two point five watt transmitter eh from on to off"
ET1AOB_0 - "change two and a half watt transmitter eh from on to off"
ET1AOB_1 - "set ee tea one eh oh bee to on"
ET1AOB_0 - "set ee tea one eh oh bee to off"
ET1AOB_1 - "set two point five watt transmitter eh to on"
ET1AOB_1 - "set two and a half watt transmitter eh to on"
ET1AOB_0 - "set two point five watt transmitter eh to off"
ET1AOB_0 - "set two and a half watt transmitter eh to off"
ET1AOB_1 - "turn ee tea one eh oh bee on"
ET1AOB_0 - "turn ee tea one eh oh bee off"
ET1AOB_1 - "turn two point five watt transmitter eh on"
ET1AOB_1 - "turn two and a half watt transmitter eh on"
ET1AOB_0 - "turn two point five watt transmitter eh off"
ET1AOB_0 - "turn two and a half watt transmitter eh off"
ET1AOB_1 - "turn on ee tea one eh oh bee"
ET1AOB_0 - "turn off ee tea one eh oh bee"
ET1AOB_1 - "turn on two point five watt transmitter eh"
ET1AOB_1 - "turn on two and a half watt transmitter eh"
ET1AOB_0 - "turn off two point five watt transmitter eh"
ET1AOB_0 - "turn off two and a half watt transmitter eh"

ET1BOB
ET1BOB_1 - "change ee tea one bee oh bee from off to on"

ET1BOB_0 - "change ee tea one bee oh bee from on to off"
ET1BOB_1 - "change two point five watt transmitter bee from off to on"
ET1BOB_1 - "change two and a half watt transmitter bee from off to on"
ET1BOB_0 - "change two point five watt transmitter bee from on to off"
ET1BOB_0 - "change two and a half watt transmitter bee from on to off"
ET1BOB_1 - "set ee tea one bee oh bee to on"
ET1BOB_0 - "set ee tea one bee oh bee to off"
ET1BOB_1 - "set two point five watt transmitter bee to on"
ET1BOB_1 - "set two and a half watt transmitter bee to on"
ET1BOB_0 - "set two point five watt transmitter bee to off"
ET1BOB_0 - "set two and a half watt transmitter bee to off"
ET1BOB_1 - "turn ee tea one bee oh bee on"
ET1BOB_0 - "turn ee tea one bee oh bee off"
ET1BOB_1 - "turn two point five watt transmitter bee on"
ET1BOB_1 - "turn two and a half watt transmitter bee on"
ET1BOB_0 - "turn two point five watt transmitter bee off"
ET1BOB_0 - "turntwo and a half watt transmitter bee off"
ET1BOB_1 - "turn on ee tea one bee oh bee"
ET1BOB_0 - "turn off ee tea one bee oh bee"
ET1BOB_1 - "turn on two point five watt transmitter bee"
ET1BOB_1 - "turn on two and a half watt transmitter bee"
ET1BOB_0 - "turn off two point five watt transmitter bee"
ET1BOB_0 - "turn off two and a half watt transmitter bee"

<u>ET1BAB</u>
ET1BAB_1 - "change ee tea one bee eh bee from safe to arm"
ET1BAB_0 - "change ee tea one bee eh bee from arm to safe"
ET1BAB_1 - "change two point five watt transmitter bee from safe to arm"
ET1BAB_1 - "change two and a half watt transmitter bee from safe to arm"
ET1BAB_0 - "change two point five watt transmitter bee from arm to safe"
ET1BAB_0 - "change two and a half watt transmitter bee from arm to safe"
ET1BAB_1 - "set ee tea one bee eh bee to arm"
ET1BAB_0 - "set ee tea one bee eh bee to safe"
ET1BAB_1 - "set two point five watt transmitter bee to arm"

ET1BAB_1 - "set two and a half watt transmitter bee to arm"
ET1BAB_0 - "set two point five watt transmitter bee to safe"
ET1BAB_0 - "set two and a half watt transmitter bee to safe"

ET1BEB
ET1BEB_1 - "change ee tea one bee ee bee from enabled to disabled"
ET1BEB_0 - "change ee tea one bee ee bee from disabled to enabled"
ET1BEB_1 - "change two point five watt transmitter bee from enabled to disabled"
ET1BEB_1 - "change two and a half watt transmitter bee from enabled to disabled"
ET1BEB_0 - "change two point five watt transmitter bee from disabled to enabled"
ET1BEB_0 - "change two and a half watt transmitter bee from disabled to enabled"
ET1BEB_1 - "set ee tea one bee ee bee to disabled"
ET1BEB_0 - "set ee tea one bee ee bee to enabled"
ET1BEB_1 - "set two point five watt transmitter bee to disabled"
ET1BEB_1 - "set two and a half watt transmitter bee to disabled"
ET1BEB_0 - "set two point five watt transmitter bee to enabled"
ET1BEB_0 - "set two and a half watt transmitter bee to enabled"
ET1BEB_1 - "disable ee tea one bee ee bee"
ET1BEB_0 - "enable ee tea one bee ee bee"
ET1BEB_1 - "disable two point five watt transmitter bee"
ET1BEB_1 - "disable two and a half watt transmitter bee"
ET1BEB_0 - "enable two point five watt transmitter bee"
ET1BEB_0 - "enable two and a half watt transmitter bee"

PROPULSION SUBYSTEM

ACJEAB
ACJEAB_1 - "change eh sea jay ee eh bee from enable to disable"
ACJEAB_0 - "change eh sea jay ee eh bee from disable to enable"
ACJEAB_1 - "change the eh attitude control jets from enable to disable"
ACJEAB_0 - "change the eh attitude control jets from disable to enable"
ACJEAB_1 - "set eh sea jay ee eh bee to disable"

ACJEAB_0 - "set eh sea jay ee eh bee to enable"
ACJEAB_1 - "set the eh attitude control jets to disable"
ACJEAB_0 - "set the eh attitude control jets to enable"
ACJEAB_1 - "disable eh sea jay ee eh bee"
ACJEAB_0 - "enable eh sea jay ee eh bee"
ACJEAB_1 - "disable the eh attitude control jets"
ACJEAB_0 - "enable the eh attitude control jets"

ACJEBB
ACJEBB_1 - "change eh sea jay ee bee bee from enable to disable"
ACJEBB_0 - "change eh sea jay ee bee bee from disable to enable"
ACJEBB_1 - "change the bee attitude control jets from enable to disable"
ACJEBB_0 - "change the bee attitude control jets from disable to enable"
ACJEBB_1 - "set eh sea jay ee bee bee to disable"
ACJEBB_0 - "set eh sea jay ee bee bee to enable"
ACJEBB_1 - "set the bee attitude control jets to disable"
ACJEBB_0 - "set the bee attitude control jets to enable"
ACJEBB_1 - "disable eh sea jay ee bee bee"
ACJEBB_0 - "enable eh sea jay ee bee bee"
ACJEBB_1 - "disable the bee attitude control jets"
ACJEBB_0 - "enable the bee attitude control jets"

ACTEAB
ACTEAB_1 - "change eh sea tea ee eh bee from enable to disable"
ACTEAB_0 - "change eh sea tea ee eh bee from disable to enable"
ACTEAB_1 - "change the eh attitude control thrusters from enable to disable"
ACTEAB_0 - "change the eh attitude control thrusters from disable to enable"
ACTEAB_1 - "set eh sea tea ee eh bee to disable"
ACTEAB_0 - "set eh sea tea ee eh bee to enable"
ACTEAB_1 - "set the eh attitude control thrusters to disable"
ACTEAB_0 - "set the eh attitude control thrusters to enable"
ACTEAB_1 - "disable eh sea tea ee eh bee"
ACTEAB_0 - "enable eh sea tea ee eh bee"
ACTEAB_1 - "disable the eh attitude control thrusters"

ACTEAB_0 - "enable the eh attitude control thrusters"

ACTEBB
ACTEBB_1 - "change eh sea tea ee bee bee from enable to disable"
ACTEBB_0 - "change eh sea tea ee bee bee from disable to enable"
ACTEBB_1 - "change the bee attitude control thrusters from enable to disable"
ACTEBB_0 - "change the bee attitude control thrusters from disable to enable"
ACTEBB_1 - "set eh sea tea ee bee bee to disable"
ACTEBB_0 - "set eh sea tea ee bee bee to enable"
ACTEBB_1 - "set the bee attitude control thrusters to disable"
ACTEBB_0 - "set the bee attitude control thrusters to enable"
ACTEBB_1 - "disable eh sea tea ee bee bee"
ACTEBB_0 - "enable eh sea tea ee bee bee"
ACTEBB_1 - "disable the bee attitude control thrusters"
ACTEBB_0 - "enable the bee attitude control thrusters"

AGGAEB
AGGAEB_1 - "change eh gee gee eh ee bee from enable to disable"
AGGAEB_0 - "change eh gee gee eh ee bee from disable to enable"
AGGAEB_1 - "change gas generator control eh from enable to disable"
AGGAEB_0 - "change gas generator control eh from disable to enable"
AGGAEB_1 - "set eh gee gee eh ee bee to disable"
AGGAEB_0 - "set eh gee gee eh ee bee to enable"
AGGAEB_1 - "set gas generator control eh to disable"
AGGAEB_0 - "set gas generator control eh to enable"
AGGAEB_1 - "disable eh gee gee eh ee bee"
AGGAEB_0 - "enable eh gee gee eh ee bee"
AGGAEB_1 - "disable gas generator control eh"
AGGAEB_0 - "enable gas generator control eh"

AGGBEB
AGGBEB_1 - "change eh gee gee bee ee bee from enable to disable"
AGGBEB_0 - "change eh gee gee bee ee bee from disable to enable"
AGGBEB_1 - "change gas generator control bee from enable to disable"

AGGBEB_0 - "change gas generator control bee from disable to enable"
AGGBEB_1 - "set eh gee gee bee ee bee to disable"
AGGBEB_0 - "set eh gee gee bee ee bee to enable"
AGGBEB_1 - "set gas generator control bee to disable"
AGGBEB_0 - "set gas generator control bee to enable"
AGGBEB_1 - "disable eh gee gee bee ee bee"
AGGBEB_0 - "enable eh gee gee bee ee bee"
AGGBEB_1 - "disable gas generator control bee"
AGGBEB_0 - "enable gas generator control bee"

<u>ASJEAB</u>
ASJEAB_1 - "change eh ess jay ee eh bee from enable to disable"
ASJEAB_0 - "change eh ess jay ee eh bee from disable to enable"
ASJEAB_1 - "change the eh spin jets from enable to disable"
ASJEAB_0 - "change the eh spin jets from disable to ensable"
ASJEAB_1 - "set eh ess jay ee eh bee to disable"
ASJEAB_0 - "set eh ess jay ee eh bee to enable"
ASJEAB_1 - "set the eh spin jets to disable"
ASJEAB_0 - "set the eh spin jets to enable"
ASJEAB_1 - "disable eh ess jay ee eh bee"
ASJEAB_0 - "enable eh ess jay ee eh bee"
ASJEAB_1 - "disable the eh spin jets"
ASJEAB_0 - "enable the eh spin jets"

<u>ASJEBB</u>
ASJEBB_1 - "change eh ess jay ee bee bee from enable to disable"
ASJEBB_0 - "change eh ess jay ee bee bee from disable to enable"
ASJEBB_1 - "change the bee spin jets from enable to disable"
ASJEBB_0 - "change the bee spin jets from disable to ensable"
ASJEBB_1 - "set eh ess jay ee bee bee to disable"
ASJEBB_0 - "set eh ess jay ee bee bee to enable"
ASJEBB_1 - "set the bee spin jets to disable"
ASJEBB_0 - "set the bee spin jets to enable"
ASJEBB_1 - "disable eh ess jay ee bee bee"

ASJEBB_0 - "enable eh ess jay ee bee bee"
ASJEBB_1 - "disable the bee spin jets"
ASJEBB_0 - "enable the bee spin jets"

<u>AVVEAB</u>
AVVEAB_1 - "change eh vee vee ee eh bee from enable to disable"
AVVEAB_0 - "change eh vee vee ee eh bee from disable to enable"
AVVEAB_1 - "change the eh delta vee thruster from enable to disable"
AVVEAB_0 - "change the eh delta vee thruster from disable to enable"
AVVEAB_1 - "set eh vee vee ee eh bee to disable"
AVVEAB_0 - "set eh vee vee ee eh bee to enable"
AVVEAB_1 - "set the eh delta vee thruster to disable"
AVVEAB_0 - "set the eh delta vee thruster to enable"
AVVEAB_1 - "disable eh vee vee ee eh bee"
AVVEAB_0 - "enable eh vee vee ee eh bee"
AVVEAB_1 - "disable the eh delta vee thruster"
AVVEAB_0 - "enable the eh delta vee thruster"

<u>AVVEBB</u>
AVVEBB_1 - "change eh vee vee ee bee bee from enable to disable"
AVVEBB_0 - "change eh vee vee ee bee bee from disable to enable"
AVVEBB_1 - "change the bee delta vee thruster from enable to disable"
AVVEBB_0 - "change the bee delta vee thruster from disable to enable"
AVVEBB_1 - "set eh vee vee ee bee bee to disable"
AVVEBB_0 - "set eh vee vee ee bee bee to enable"
AVVEBB_1 - "set the bee delta vee thruster to disable"
AVVEBB_0 - "set the bee delta vee thruster to enable"
AVVEBB_1 - "disable eh vee vee ee bee bee to disable"
AVVEBB_0 - "enable eh vee vee ee bee bee to enable"
AVVEBB_1 - "disable the bee delta vee thruster"
AVVEBB_0 - "enable the bee delta vee thruster"

ELECTRICAL POWER SUBSYSTEM

EAACMB
EAACMB_1 - "change ee eh eh sea em bee from mode two to mode one"
EAACMB_0 - "change ee eh eh sea em bee from mode one to mode two"
EAACMB_0 - "change the battery eh auto charge mode from one to two"
EAACMB_1 - "change the battery eh auto charge mode from two to one"
EAACMB_1 - "set ee eh eh sea em bee to mode one"
EAACMB_0 - "set ee eh eh sea em bee to mode two"
EAACMB_0 - "set battery eh auto charge mode to two"
EAACMB_1 - "set battery eh auto charge mode to one"

EADBYB
EADBYB_1 - "change ee eh dee bee why bee from in to bye pass"
EADBYB_0 - "change ee eh dee bee why bee from bye pass to in"
EADBYB_1 - "change the battery eh diode from in to bye pass"
EADBYB_0 - "change the battery eh diode from bye pass to in"
EADBYB_1 - "set ee eh dee bee why bee to bye pass"
EADBYB_0 - "set ee eh dee bee why bee to in"
EADBYB_1 - "set the battery eh diode to bye pass"
EADBYB_0 - "set the battery eh diode to in"

EASBPB
EASBPB_1 - "change ee eh ess bee pea bee from in to bye pass"
EASBPB_0 - "change ee eh ess bee pea bee from bye pass to in"
EASBPB_1 - "change the prime array switching unit base panels from in to bye pass"
EASBPB_0 - "change the prime array switching unit base panels from bye pass to in"
EASBPB_1 - "set ee eh ess bee pea bee to bye pass"
EASBPB_0 - "set ee eh ess bee pea bee to in"
EASBPB_1 - "set the prime array switching unit base panels to bye pass"
EASBPB_0 - "set the prime array switching unit base panels to in"

EASBRB
EASBRB_1 - "change ee eh ess bee are bee from disconnect to connect"

EASBRB_0 - "change ee eh ess bee are bee from connect to disconnect"
EASBRB_1 - "change the redundant array switching unit base panel from disconnect to connect"
EASBRB_0 - "change the redundant array switching unit base panel from connect to disconnect"
EASBRB_1 - "set ee eh ess bee are bee to connect"
EASBRB_0 - "set ee eh ess bee are bee to disconnect"
EASBRB_1 - "set the redundant array switching unit base panel to connect"
EASBRB_0 - "set the redundant array switching unit base panel to disconnect"
EASBRB_1 - "connect ee eh ess bee are bee"
EASBRB_0 - "disconnect ee eh ess bee are bee"
EASBRB_1 - "connect the redundant array switching unit base panel"
EASBRB_0 - "disconnect the redundant array switching unit base panel"

<u>EBACMB</u>
EBACMB_1 - "change ee bee eh sea em bee from mode two to mode one"
EBACMB_0 - "change ee bee eh sea em bee from mode one to mode two"
EBACMB_0 - "change the battery bee auto charge mode from one to two"
EBACMB_1 - "change the battery bee auto charge mode from two to one"
EBACMB_1 - "set ee bee eh sea em bee to mode one"
EBACMB_0 - "set ee bee eh sea em bee to mode two"
EBACMB_0 - "set the battery bee auto charge mode to two"
EBACMB_1 - "set the battery bee auto charge mode to one"

<u>EBAHEB</u>
EBAHEB_1 - "change ee bee eh H ee bee from disable to enable"
EBAHEB_0 - " change ee bee eh H ee bee from enable to disable"
EBAHEB_1 - "change the battery eh heater power from disable to enable"
EBAHEB_0 - "change the battery eh heater power from enable to disable"
EBAHEB_1 - "set ee bee eh H ee bee to enable"
EBAHEB_0 - "set ee bee eh H ee bee to disable"
EBAHEB_1 - "set the battery eh heater power to enable"
EBAHEB_0 - "set the battery eh heater power to disable"
EBAHEB_1 - "enable ee bee eh H ee bee"
EBAHEB_0 - "disable ee bee eh H ee bee"
EBAHEB_1 - "enable the battery eh heater power"

EBAHEB_0 - "disable the battery eh heater power"

EBAK1B
EBAK1B_1 - "change ee bee eh kay one bee from full to trickle"
EBAK1B_0 - "change ee bee eh kay one bee from trickle to full"
EBAK1B_1 - "change the battery eh kay one position from full to trickle"
EBAK1B_0 - "change the battery eh kay one position from trickle to full"
EBAK1B_1 - "set ee bee eh kay one bee to trickle"
EBAK1B_0 - "set ee bee eh kay one bee to full"
EBAK1B_1 - "set the battery eh kay one position to trickle"
EBAK1B_0 - "set the battery eh kay one position to full"

EBAK2B
EBAK2B_1 - "change ee bee eh kay two bee from connect to disconnect"
EBAK2B_0 - "change ee bee eh kay two bee from disconnect to connect"
EBAK2B_1 - "change the battery eh kay two relay position from connect to disconnect"
EBAK2B_0 - "change the battery eh kay two relay position from disconnect to connect"
EBAK2B_1 - "set ee bee eh kay two bee to disconnect"
EBAK2B_0 - "set ee bee eh kay two bee to connect"
EBAK2B_1 - "set the battery eh kay two relay position to disconnect"
EBAK2B_0 - "set the battery eh kay two relay position to connect"
EBAK2B_1 - "disconnect ee bee eh kay two bee"
EBAK2B_0 - "connect ee bee eh kay two bee"
EBAK2B_1 - "disconnect the battery eh kay two relay"
EBAK2B_0 - "connect the battery eh kay two relay"

EBAK3B
EBAK3B_1 - "change ee bee eh kay three bee from manual to automatic"
EBAK3B_1 - "change ee bee eh kay three bee from manual to auto"
EBAK3B_0 - "change ee bee eh kay three bee from automatic to manual"
EBAK3B_0 - "change ee bee eh kay three bee from auto to manual"
EBAK3B_1 - "change battery eh kay three relay position from manual to automatic"
EBAK3B_1 - "change battery eh kay three relay position from manual to auto"
EBAK3B_0 - "change battery eh kay three relay position from automatic to manual"

EBAK3B_0 - "change battery eh kay three relay position from auto to manual"
EBAK3B_1 - "set ee bee eh kay three bee to automatic"
EBAK3B_1 - "set ee bee eh kay three bee to auto"
EBAK3B_0 - "set ee bee eh kay three bee to manual"
EBAK3B_1 - "set battery eh kay three relay position to automatic"
EBAK3B_1 - "set battery eh kay three relay position to auto"
EBAK3B_0 - "set battery eh kay three relay position to manual"

<u>EBARDB</u>
EBARDB_1 - "change ee bee eh are dee bee from recondition to open"
EBARDB_0 - "change ee bee eh are dee bee from open to recondition"
EBARDB_1 - "change battery eh from recondition to open"
EBARDB_0 - "change battery eh from open to recondition"
EBARDB_1 - "set ee bee eh are dee bee to open"
EBARDB_0 - "set ee bee eh are dee bee to recondition"
EBARDB_1 - "set battery eh to open"
EBARDB_0 - "set battery eh to recondition"

<u>EBARTB</u>
EBARTB_1 - "change ee bee eh are tea bee from manual to automatic"
EBARTB_1 - "change ee bee eh are tea bee from manual to auto"
EBARTB_0 - "change ee bee eh are tea bee from automatic to manual"
EBARTB_0 - "change ee bee eh are tea bee from auto to manual"
EBARTB_1 - "change the battery eh recondition termination from manual to automatic"
EBARTB_1 - "change the battery eh recondition termination from manual to auto"
EBARTB_0 - "change the battery eh recondition termination from automatic to manual"
EBARTB_0 - "change the battery eh recondition termination from auto to manual"
EBARTB_1 - "set ee bee eh are tea bee to automatic"
EBARTB_1 - "set ee bee eh are tea bee to auto"
EBARTB_0 - "set ee bee eh are tea bee to manual"
EBARTB_1 - "set the battery eh recondition termination to automatic"
EBARTB_1 - "set the battery eh recondition termination to auto"
EBARTB_0 - "set the battery eh recondition termination to manual"

EBBHEB

EBBHEB_1 - "change ee bee bee H ee bee from disable to enable"
EBBHEB_0 - "change ee bee bee H ee bee from enable to disable"
EBBHEB_1 - "change the battery bee heater power from disable to enable"
EBBHEB_0 - "change the battery bee heater power from enable to disable"
EBBHEB_1 - "set ee bee bee H ee bee to enable"
EBBHEB_0 - "set ee bee bee H ee bee to disable"
EBBHEB_1 - "set battery bee heater power to enable"
EBBHEB_0 - "set battery bee heater power to disable"
EBBHEB_1 - "enable ee bee bee H ee bee"
EBBHEB_0 - "disable ee bee bee H ee bee"
EBBHEB_1 - "enable battery bee heater power"
EBBHEB_0 - "disable battery bee heater power"

EBBK1B

EBBK1B_1 - "change ee bee bee kay one bee from full to trickle"
EBBK1B_0 - "change ee bee bee kay one bee from trickle to full"
EBBK1B_1 - "change the battery bee kay one relay position from full to trickle"
EBBK1B_0 - "change the battery bee kay one relay position from trickle to full"
EBBK1B_1 - "set ee bee bee kay one bee to trickle"
EBBK1B_0 - "set ee bee bee kay one bee to full"
EBBK1B_1 - "set the battery bee kay one relay position to trickle"
EBBK1B_0 - "set the battery bee kay one relay position to full"

EBBK2B

EBBK2B_1 - "change ee bee bee kay two bee from connect to disconnect"
EBBK2B_0 - "change ee bee bee kay two bee from disconnect to connect"
EBBK2B_1 - "change the battery bee kay two relay position from connect to disconnect"
EBBK2B_0 - "change the battery bee kay two relay position from disconnect to connect"
EBBK2B_1 - "set bee bee kay two bee to disconnect"
EBBK2B_0 - "set bee bee kay two bee to connect"
EBBK2B_1 - "set the battery bee kay two relay position to disconnect"
EBBK2B_0 - "set the battery bee kay two relay position to connect"
EBBK2B_1 - "disconnect bee bee kay two bee"

EBBK2B_0 - "connect bee bee kay two bee"
EBBK2B_1 - "disconnect the battery bee kay two relay"
EBBK2B_0 - "connect the battery bee kay two relay"

EBBK3B
EBBK3B_1 - "change ee bee bee kay three bee from manual to automatic"
EBBK3B_1 - "change ee bee bee kay three bee from manual to auto"
EBBK3B_0 - "change ee bee bee kay three bee from automatic to manual"
EBBK3B_0 - "change ee bee bee kay three bee from auto to manual"
EBBK3B_1 - "change battery bee kay three relay position from manual to automatic"
EBBK3B_1 - "change battery bee kay three relay position from manual to auto"
EBBK3B_0 - "change battery bee kay three relay position from automatic to manual"
EBBK3B_0 - "change battery bee kay three relay position from auto to manual"
EBBK3B_1 - "set ee bee bee kay three bee to automatic"
EBBK3B_1 - "set ee bee bee kay three bee to auto"
EBBK3B_0 - "set ee bee bee kay three bee to manual"
EBBK3B_1 - "set battery bee kay three relay position to automatic"
EBBK3B_1 - "set battery bee kay three relay position to auto"
EBBK3B_0 - "set battery bee kay three relay position to manual"

EBBRDB
EBBRDB_1 - "change ee bee bee are dee bee from recondition to open"
EBBRDB_0 - "change ee bee bee are dee bee from open to recondition"
EBBRDB_1 - "change battery bee from recondition to open"
EBBRDB_0 - "change battery bee from open to recondition"
EBBRDB_1 - "set ee bee bee are dee bee to open"
EBBRDB_0 - "set ee bee bee are dee bee to recondition"
EBBRDB_1 - "set battery bee to open"
EBBRDB_0 - "set battery bee to recondition"

EBBRTB
EBBRTB_1 - "change ee bee bee are tea bee from manual to automatic"
EBBRTB_1 - "change ee bee bee are tea bee from manual to auto"
EBBRTB_0 - "change ee bee bee are tea bee from automatic to manual"

EBBRTB_0 - "change ee bee bee are tea bee from auto to manual"
EBBRTB_1 - "change the battery bee recondition termination from manual to automatic"
EBBRTB_1 - "change the battery bee recondition termination from manual to auto"
EBBRTB_0 - "change the battery bee recondition termination from automatic to manual"
EBBRTB_0 - "change the battery bee recondition termination from auto to manual"
EBBRTB_1 - "set ee bee bee are tea bee to automatic"
EBBRTB_1 - "set ee bee bee are tea bee to auto"
EBBRTB_0 - "set ee bee bee are tea bee to manual"
EBBRTB_1 - "set the battery bee recondition termination to automatic"
EBBRTB_1 - "set the battery bee recondition termination to auto"
EBBRTB_0 - "set the battery bee recondition termination to manual"

EBCHEB
EBCHEB_1 - "change ee bee sea H ee bee from disable to enable"
EBCHEB_0 - "change ee bee sea H ee bee from enable to disable"
EBCHEB_1 - "change the battery sea heater power from disable to enable"
EBCHEB_0 - "change the battery sea heater power from enable to disable"
EBCHEB_1 - "set ee bee sea H ee bee to enable"
EBCHEB_0 - "set ee bee sea H ee bee to disable"
EBCHEB_1 - "set the battery sea heater power to enable"
EBCHEB_0 - "set the battery sea heater power to disable"
EBCHEB_1 - "enable ee bee sea H ee bee"
EBCHEB_0 - "disable ee bee sea H ee bee"
EBCHEB_1 - "enable the battery sea heater power"
EBCHEB_0 - "disable the battery sea heater power"

EBCK1B
EBCK1B_1 - "change ee bee sea kay one bee from full to trickle"
EBCK1B_0 - "change ee bee sea kay one bee from trickle to full"
EBCK1B_1 - "change the battery sea kay one position from full to trickle"
EBCK1B_0 - "change the battery sea kay one position from trickle to full"
EBCK1B_1 - "set ee bee sea kay one bee to trickle"
EBCK1B_0 - "set ee bee sea kay one bee to full"
EBCK1B_1 - "set the battery sea kay one position to trickle"

EBCK1B_0 - "set the battery sea kay one position to full"

EBCK2B

EBCK2B_1 - "change ee bee sea kay two bee from connect to disconnect"
EBCK2B_0 - "change ee bee sea kay two bee from disconnect to connect"
EBCK2B_1 - "change the battery sea kay two relay position from connect to disconnect"
EBCK2B_0 - "change the battery sea kay two relay position from disconnect to connect"
EBCK2B_1 - "set ee bee sea kay two bee to disconnect"
EBCK2B_0 - "set ee bee sea kay two bee to connect"
EBCK2B_1 - "set the battery sea kay two relay position to disconnect"
EBCK2B_0 - "set the battery sea kay two relay position to connect"
EBCK2B_1 - "disconnect ee bee sea kay two bee"
EBCK2B_0 - "connect ee bee sea kay two bee"
EBCK2B_1 - "disconnect the battery sea kay two relay"
EBCK2B_0 - "connect the battery sea kay two relay"

EBCK3B

EBCK3B_1 - "change ee bee sea kay three bee from manual to automatic"
EBCK3B_1 - "change ee bee sea kay three bee from manual to auto"
EBCK3B_0 - "change ee bee sea kay three bee from automatic to manual"
EBCK3B_0 - "change ee bee sea kay three bee from auto to manual"
EBCK3B_1 - "change battery sea kay three relay position from manual to automatic"
EBCK3B_1 - "change battery sea kay three relay position from manual to auto"
EBCK3B_0 - "change battery sea kay three relay position from automatic to manual"
EBCK3B_0 - "change battery sea kay three relay position from auto to manual"
EBCK3B_1 - "set ee bee sea kay three bee to automatic"
EBCK3B_1 - "set ee bee sea kay three bee to auto"
EBCK3B_0 - "set ee bee sea kay three bee to manual"
EBCK3B_1 - "set battery sea kay three relay position to automatic"
EBCK3B_1 - "set battery sea kay three relay position to auto"
EBCK3B_0 - "set battery sea kay three relay position to manual"

EBCRDB

EBCRDB_1 - "change ee bee sea are dee bee from recondition to open"

EBCRDB_0 - "change ee bee sea are dee bee from open to recondition"
EBCRDB_1 - "change battery sea from recondition to open"
EBCRDB_0 - "change battery sea from open to recondition"
EBCRDB_1 - "set ee bee sea are dee bee to open"
EBCRDB_0 - "set ee bee sea are dee bee to recondition"
EBCRDB_1 - "set battery sea to open"
EBCRDB_0 - "set battery sea to recondition"

<u>EBCRTB</u>
EBCRTB_1 - "change ee bee sea are tea bee from manual to automatic"
EBCRTB_1 - "change ee bee sea are tea bee from manual to auto"
EBCRTB_0 - "change ee bee sea are tea bee from automatic to manual"
EBCRTB_0 - "change ee bee sea are tea bee from auto to manual"
EBCRTB_1 - "change the battery sea recondition termination from manual to automatic"
EBCRTB_1 - "change the battery sea recondition termination from manual to auto"
EBCRTB_0 - "change the battery sea recondition termination from automatic to manual"
EBCRTB_0 - "change the battery sea recondition termination from auto to manual"
EBCRTB_1 - "set ee bee sea are tea bee to automatic"
EBCRTB_1 - "set ee bee sea are tea bee to auto"
EBCRTB_0 - "set ee bee sea are tea bee to manual"
EBCRTB_1 - "set the battery sea recondition termination to automatic"
EBCRTB_1 - "set the battery sea recondition termination to auto"
EBCRTB_0 - "set the battery sea recondition termination to manual"

<u>EBDBYB</u>
EBDBYB_1 - "change ee bee dee bee why bee from in to bye pass"
EBDBYB_0 - "change ee bee dee bee why bee from bye pass to in"
EBDBYB_1 - "change the battery bee diode from in to bye pass"
EBDBYB_0 - "change the battery bee diode from bye pass to in"
EBDBYB_1 - "set ee bee dee bee why bee to bye pass"
EBDBYB_0 - "set ee bee dee bee why bee to in"
EBDBYB_1 - "set the battery bee diode to bye pass"
EBDBYB_0 - "set the battery bee diode to in"

<u>EBVLSB</u>
EBVLSB_1 - "change ee bee vee ell ess bee from high to low"
EBVLSB_0 - "change ee bee vee ell ess bee from low to high"
EBVLSB_1 - "change the bus voltage limiter from high to low"
EBVLSB_0 - "change the bus voltage limiter from low to high"
EBVLSB_1 - "set ee bee vee ell ess bee to low"
EBVLSB_0 - "set ee bee vee ell ess bee to high"
EBVLSB_1 - "set the bus voltage limiter to low"
EBVLSB_0 - "set the bus voltage limiter to high"

<u>ECACMB</u>
ECACMB_1 - "change ee sea eh sea em bee from mode two to mode one"
ECACMB_0 - "change ee sea eh sea em bee from mode one to mode two"
ECACMB_0 - "change the battery sea auto change mode from one to two"
ECACMB_1 - "change the battery sea auto change mode from two to one"
ECACMB_1 - "set ee sea eh sea em bee to mode one"
ECACMB_0 - "set ee sea eh sea em bee to mode two"
ECACMB_0 - "set the battery sea auto change mode to two"
ECACMB_1 - "set the battery sea auto change mode to one"

<u>ECDBYB</u>
ECDBYB_1 - "change ee sea dee bee why bee from in to bye pass"
ECDBYB_0 - "change ee sea dee bee why bee from bye pass to in"
ECDBYB_1 - "change the battery sea diode from in to bye pass"
ECDBYB_0 - "change the battery sea diode from bye pass to in"
ECDBYB_1 - "set ee sea dee bee why bee to bye pass"
ECDBYB_0 - "set ee sea dee bee why bee to in"
ECDBYB_1 - "set the battery sea diode to bye pass"
ECDBYB_0 - "set the battery sea diode to in"
ECDBYB_1 - "set ee sea dee bee why bee to bye pass"
ECDBYB_0 - "set ee sea dee bee why bee to in"
ECDBYB_1 - "set the battery sea diode to bye pass"
ECDBYB_0 - "set the battery sea diode to in"
ECDBYB_1 - "set ee sea dee bee why bee to bye pass"

ECDBYB_0 - "set ee sea dee bee why bee to in"
ECDBYB_1 - "set the battery sea diode to bye pass"
ECDBYB_0 - "set the battery sea diode to in"
ECDBYB_0 - "disable ee sea dee bee why bee"
ECDBYB_1 - "enable ee sea dee bee why bee"
ECDBYB_0 - "disable the battery sea diode bypass"
ECDBYB_1 - "enable the battery sea diode bypass"

<u>EDUVSB</u>
EDUVSB_1 - "change ee dee you vee ess bee from overide to normal"
EDUVSB_1 - "change ee dee you vee ess bee from overide to norm"
EDUVSB_0 - "change ee dee you vee ess bee from normal to overide"
EDUVSB_0 - "change ee dee you vee ess bee from norm to overide"
EDUVSB_1 - "change the undervoltage status from overide to normal"
EDUVSB_1 - "change the undervoltage status from overide to norm"
EDUVSB_0 - "change the undervoltage status from normal to overide"
EDUVSB_0 - "change the undervoltage status from norm to overide"
EDUVSB_1 - "set ee dee you vee ess bee to normal"
EDUVSB_1 - "set ee dee you vee ess bee to norm"
EDUVSB_0 - "set ee dee you vee ess bee to overide"
EDUVSB_1 - "set the undervoltage status to normal"
EDUVSB_1 - "set the undervoltage status to norm"
EDUVSB_0 - "set the undervoltage status to overide"

<u>ESP1PB</u>
ESP1PB_1 - "change ee ess pea one pea bee from disconnect to connect"
ESP1PB_0 - "change ee ess pea one pea bee from connect to disconnect"
ESP1PB_1 - "change increment one on the prime array switching paddle increment from disconnect to connect"
ESP1PB_0 - "change increment one on the prime array switching paddle increment from connect to disconnect"
ESP1PB_1 - "set ee ess pea one pea bee to connect"
ESP1PB_0 - "set ee ess pea one pea bee to disconnect"
ESP1PB_1 - "set increment one on the prime array switching paddle increment to connect"
ESP1PB_0 - "set increment one on the prime array switching paddle increment to disconnect"

ESP1RB

ESP1RB_1 - "change ee ess pea one are bee from disconnect to connect"
ESP1RB_0 - "change ee ess pea one are bee from connect to disconnect"
ESP1RB_1 - "change increment one on the redundant array switching paddle increment from disconnect to connect"
ESP1RB_0 - "change increment one on the redundant array switching paddle increment from connect to disconnect"
ESP1RB_1 - "set ee ess pea one are bee to connect"
ESP1RB_0 - "set ee ess pea one are bee to disconnect"
ESP1RB_1 - "set increment one on the redundant array switching paddle increment to connect"
ESP1RB_0 - "set increment one on the redundant array switching paddle increment to disconnect"

ESP2PB

ESP2PB_1 - "change ee ess pea two pea bee from disconnect to connect"
ESP2PB_0 - "change ee ess pea two pea bee from connect to disconnect"
ESP2PB_1 - "change increment two on the prime array switching paddle increment from disconnect to connect"
ESP2PB_0 - "change increment two on the prime array switching paddle increment from connect to disconnect"
ESP2PB_1 - "set ee ess pea two pea bee to connect"
ESP2PB_0 - "set ee ess pea two pea bee to disconnect"
ESP2PB_1 - "set increment two on the prime array switching paddle increment to connect"
ESP2PB_0 - "set increment two on the prime array switching paddle increment to disconnect"

ESP2RB

ESP2RB_1 - "change ee ess pea two are bee from disconnect to connect"
ESP2RB_0 - "change ee ess pea two are bee from connect to disconnect"
ESP2RB_1 - "change increment two on the redundant array switching paddle increment from disconnect to connect"
ESP2RB_0 - "change increment two on the redundant array switching paddle increment from connect to disconnect"
ESP2RB_1 - "set ee ess pea two are bee to connect"
ESP2RB_0 - "set ee ess pea two are bee to disconnect"
ESP2RB_1 - "set increment two on the redundant array switching paddle increment to connect"
ESP2RB_0 - "set increment two on the redundant array switching paddle increment to disconnect"

ESP3PB

ESP3PB_1 - "change ee ess pea three pea bee from disconnect to connect"
ESP3PB_0 - "change ee ess pea three pea bee from connect to disconnect"
ESP3PB_1 - "change increment three on the prime array switching paddle increment from disconnect to connect"

ESP1RB
ESP1RB_1 - "change ee ess pea one are bee from disconnect to connect"
ESP1RB_0 - "change ee ess pea one are bee from connect to disconnect"
ESP1RB_1 - "change increment one on the redundant array switching paddle increment from disconnect to connect"
ESP1RB_0 - "change increment one on the redundant array switching paddle increment from connect to disconnect"
ESP1RB_1 - "set ee ess pea one are bee to connect"
ESP1RB_0 - "set ee ess pea one are bee to disconnect"
ESP1RB_1 - "set increment one on the redundant array switching paddle increment to connect"
ESP1RB_0 - "set increment one on the redundant array switching paddle increment to disconnect"

ESP2PB
ESP2PB_1 - "change ee ess pea two pea bee from disconnect to connect"
ESP2PB_0 - "change ee ess pea two pea bee from connect to disconnect"
ESP2PB_1 - "change increment two on the prime array switching paddle increment from disconnect to connect"
ESP2PB_0 - "change increment two on the prime array switching paddle increment from connect to disconnect"
ESP2PB_1 - "set ee ess pea two pea bee to connect"
ESP2PB_0 - "set ee ess pea two pea bee to disconnect"
ESP2PB_1 - "set increment two on the prime array switching paddle increment to connect"
ESP2PB_0 - "set increment two on the prime array switching paddle increment to disconnect"

ESP2RB
ESP2RB_1 - "change ee ess pea two are bee from disconnect to connect"
ESP2RB_0 - "change ee ess pea two are bee from connect to disconnect"
ESP2RB_1 - "change increment two on the redundant array switching paddle increment from disconnect to connect"
ESP2RB_0 - "change increment two on the redundant array switching paddle increment from connect to disconnect"
ESP2RB_1 - "set ee ess pea two are bee to connect"
ESP2RB_0 - "set ee ess pea two are bee to disconnect"
ESP2RB_1 - "set increment two on the redundant array switching paddle increment to connect"
ESP2RB_0 - "set increment two on the redundant array switching paddle increment to disconnect"

ESP3PB
ESP3PB_1 - "change ee ess pea three pea bee from disconnect to connect"
ESP3PB_0 - "change ee ess pea three pea bee from connect to disconnect"
ESP3PB_1 - "change increment three on the prime array switching paddle increment from disconnect to connect"

379

ESP4RB_0 - "set increment four on the redundant array switching paddle increment to disconnect"

CAUTION LIST DISPLAY

ShowCautionList - "Show me the list of System Cautions"
ShowCautionList - "Show the list of System Cautions"
ShowCautionList - "Show me the list of Cautions"
ShowCautionList - "Show the list of Cautions"
ShowCautionList - "Display the list of System Cautions"
ShowCautionList - "Display the list of Cautions"
ShowCautionList - "What is the list of System Cautions"
ShowCautionList - "What is the list of Cautions"

CAUTION LIST REMOVE

RemoveCautionList - "Remove caution list"

WARNING LIST DISPLAY

ShowWarningList - "Show me the list of System Warnings"
ShowWarningList - "Show the list of System Warnings"
ShowWarningList - "Show me the list of Warnings"
ShowWarningList - "Show the list of Warnings"
ShowWarningList - "Display the list of System Warnings"
ShowWarningList - "Display the list of Warnings"
ShowWarningList - "What is the list of System Warnings"
ShowWarningList - "What is the list of Warnings"

WARNINGS LIST REMOVE

RemoveWarningList - "Remove warnings list"

LINK 2 COMMUNICATIONS FLOW DISPLAY

Link2CommunicationsShow - "Show me Link 2 Communications flow"

Link2CommunicationsRemove - "Remove Link 2 Communications flow"

LINK 1 COMMUNICATIONS FLOW DISPLAY

Link1CommunicationsShow - "Show me Link 1 Communications flow"
Link1CommunicationsRemove - "Remove Link 1 Communications flow"

ELECTRICAL POWER FLOW DISPLAY

ElectricalPowerShow - "Show me Electrical Power flow"
ElectricalPowerRemove - "Remove Electrical Power flow"

PASS PLAN 1 DISPLAY

PassPlanOneShow - "Show me pass plan one"
PassPlanOneShow - "Show me pass plan to set sea double you three pea bee to two point five watts"
PassPlanOneShow - "Show me pass plan to set sea double you three pea bee to two and a half watts"
PassPlanOneShow - "Show me pass plan to set the the link one eh side to two point five watts"
PassPlanOneShow - "Show me pass plan to set the the link one eh side to two and a half watts"
PassPlanOneRemove - "Remove pass plan one"

PASS PLAN 2 DISPLAY

// this is from the COBRA Training 1 scenario
PassPlanTwoShow - "Show me pass plan two"
PassPlanTwoShow - "Show me the pass plan to set the link two eh side to two point five watts"
PassPlanTwoShow - "Show me the pass plan to set the link two eh side to two and a half watts"
PassPlanTwoShow - "Show me the pass plan for an eh side power amp failure"
PassPlanTwoShow - "Show me the pass plan for an eh side power amplifier failure"
PassPlanTwoShow - "Show me the pass plan to turn the eh side amp and transmitter off and the bee side on"
PassPlanTwoRemove - "Remove pass plan two"

PASS PLAN 3 DISPLAY

//this is from the COBRA Training 2 scenario
PassPlanThreeShow - "Show me pass plan three"

PassPlanThreeShow - "Show me the pass for an eh side two point five watt transmitter failure"
PassPlanThreeShow - "Show me the pass for an eh side two and a half watt transmitter failure"
PassPlanThreeRemove - "Remove pass plan three"

PASS PLAN 4 DISPLAY

// this is from the COBRA Training 3 scenario
PassPlanFourShow - "Show me pass plan four
PassPlanFourShow - "Show me the pass for an eh side twenty watt transmitter failure"
PassPlanFourRemove - "Remove pass plan four"

PASS PLAN 5 DISPLAY

// this is from the COBRA Training 4 scenario
PassPlanFiveShow - "Show me pass plan five"
PassPlanFiveShow - "Show me the pass plan for a bee side transmitter failure"

PassPlanFiveRemove - "Remove pass plan five"

PASS PLAN 6 DISPLAY

// this is from the COBRA Training 5 scenario
PassPlanSixShow - "Show me pass plan six"
PassPlanSixShow - "Show me the pass plan for an eh side data rate unit failure"
PassPlanSixRemove - "Remove pass plan six"

PASS PLAN 7 DISPLAY

// this is from the COBRA Training 6 scenario
PassPlanSevenShow - "Show me pass plan seven"
PassPlanSevenShow - "Show me the pass plan to change the data rate from 2048 kay to 1024 kay"
PassPlanSevenShow - "Show me the pass plan to change the data rate from two thousand fourty eight kay to one thousand twenty four kay"
PassPlanSevenShow - "Show me the pass plan to change the data rate from two thousand fourty eight to one thousand twenty four"
PassPlanSevenShow - "Show me the pass plan to change the data rate from two oh four eight kay to one oh two four kay"
PassPlanSevenShow - "Show me the pass plan to change the data rate from two oh four eight to one oh two four"

PassPlanSevenShow - "Show me the pass plan to change the data rate from two thousand fourty eight kay to ten twenty four kay"
PassPlanSevenShow - "Show me the pass plan to change the data rate from two thousand fourty eight to ten twenty four"
PassPlanSevenRemove - "Remove pass plan seven"

PASS PLAN 8 DISPLAY

// this is from the COBRA Training 7 scenario
PassPlanEightShow - "Show me pass plan eight"
PassPlanEightShow - "Show me the pass plan for an eh side cryptograph failure"
PassPlanEightShow - "Show me the pass plan for an eh side crypto failure"
PassPlanEightRemove - "Remove pass plan eight"

PASS PLAN 9 DISPLAY

// this is from the COBRA Training 8 scenario
PassPlanNineShow - "Show me pass plan nine"
PassPlanNineShow - "Show me the pass plan for an eh side power amplifier overheating problem"
PassPlanNineShow - "Show me the pass plan for an eh side power amp overheating problem"
PassPlanNineShow - "Show me the pass plan for an eh side power amp heating problem"
PassPlanNineShow - "Show me the pass plan for an eh side heating problem"
PassPlanNineShow - "Show me the pass plan for an eh side power amp over temp problem"
PassPlanNineShow - "Show me the pass plan for an eh side over temp problem"
PassPlanNineRemove - "Remove pass plan nine"

Strip Chart for variables

// These are used to display static data on a strip chart.

CPA1AT

// In the case of variable CPA1AT, the temperature of power amplifier A in the Link 1 Comms subsystem,
// the data show the temperature starting in the normal region, and then gradually climbing through the
// "yellow" and into the "red" regions. The temperature then stabilizes in the red region. This pattern
// is used in COBRA Training Scenario 8. This scenario was provided by Chad Oster of CERES.
vs_StripChart - "Show me the strip chart"

vs_StripChart - "Show me the history of sea pea eh one eh tea"
vs_StripChart - "Show me the temperatures of sea pea eh one eh tea over the past month"
vs_StripChart - "Show me the values of sea pea eh one eh tea over the past month"
vs_StripChart - "Show me the history of power amplifier eh temperatures"
vs_StripChart - "Show me the temperatures of power amplifier eh over the past month"
vs_StripChart - "Show me the values of power amplifier eh over the past month"

Strip Chart 2

vs_StripChart_1 - "Show me the new strip chart"

Appendix 3 – List of Satellite Variables Simulated

Table of Contents

Background

The satellite simulation implemented in the test bed represents a generic DSCS satellite. Four subsystems are simulated. These subsystems are:

- Link 1 Communications
- Link 2 Communications
- Propulsion
- Electrical

The tables below describe the variables simulated in the test bed. These variables are grouped by subsystem. Each table contains the name of the variable and a description of the variable. For continuous variables, the tables also contain the values that define the nominal, warning, and caution ranges. For discrete variables, the two states are defined.

These variables and values were provided by CERES for use in this program.

Link 1 Communications Subsystem

Variable Name	Description	State Assignments	Critical Low	Nominal Low	Nominal High	Critical High	Nominal
CCT1AV	2.5w XMTR A Output Voltage		0.00	0.00	2.33	2.39	1.17
CCT1AW	RF Power Monitor 2.5w XMTR - A		32.48	32.48	36.61	37.15	34.55
CCT1BV	2.5w XMTR B Output Voltage		0.00	0.00	0.20	0.20	0.10
CCT1BW	RF Power Monitor 2.5w XMTR - B		33.10	33.10	35.00	35.00	34.05
CDCAMB	Dual Error Coder-A Mode Status	0=Bypass 1=Code					
CDCBMB	Dual Error Coder-B Mode Status	0=Bypass 1=Code					
CECA5V	: +5 VDC Dual Error Coder - A		4.64	4.72	4.96	5.08	4.84
CECB5V	: +5 VDC Dual Error Coder - B		0.00	0.00	1.20	1.20	0.60
CPA1AT	Solid State Power Amplifier Temp - A		22.24	22.24	47.75	52.86	35.00
CPA1AV	Solid State Power Amplifier Voltage - A		0.00	0.00	0.20	0.20	0.10
CPA1AW	Solid State Power Amplifier RF Power Monitor - A		40.00	40.00	44.00	44.00	42.00
CPA1BT	Solid State Power Amplifier Temp - B		22.24	22.24	49.91	54.78	36.08
CPA1BV	Solid State Power Amplifier Voltage - B		0.00	0.00	0.20	0.20	0.10
CPA1BW	Solid State Power Amplifier RF Power Monitor - B		40.00	40.00	44.00	44.00	42.00
CSW1PB	RF Switch 1 Position XMTR-A	0=2.5w 1=20w					
CSW2PB	RF Switch 2 Position XMTR-A	0=2.5w 1=20w					
CSW3PB	RF Switch 3 Link 1 Channel Select	0=BCH 1=ACH					
CSW4PB	RF Switch 4 Position XMTR-B	0=2.5w 1=20w					
CSW5PB	RF Switch 5 Position XMTR-B	0=2.5w 1=20w					
CT1AMT	Temp Monitor 2.5w XMTR A		10.00	40.00	85.00	110.00	62.50
CT1BMT	Temp Monitor 2.5w XMTR B		10.00	40.00	85.00	110.00	62.50
E1ASOB	Link 1A RF SW INHIBIT Override	0=OVRD 1=NORM					
E1BSOB	Link 1B RF SW INHIBIT Override	0=OVRD 1=NORM					
EDC1AB	Data Control Link 1A	0=ENCR 1=BYPASS					
EDECXB	Duel Error Coder Status (X = A or B)	0=OFF 1=ON					
EKG1AB	Link 1 KGX-28A-A Status	0=OFF 1=ON					
EKG1BB	Link 1 KGX-28A-B Status	0=OFF 1=ON					
EP1AHB	Solid State Power Amplifier Heater Status - A	0=Dis 1=Ena					
EP1XSB	Solid State Power Amplifier Status (X = A or B)	0=OFF 1=ON					
EP1BHB	Solid State Power Amplifier Heater Status - B	0=Dis 1=Ena					
ET1AAB	2.5w XMTR-A Arm	0=Safe 1=Arm					
ET1AEB	2.5w XMTR-A Enable	0=Ena 1=Dis					
ET1XOB	2.5w XMTR Status (X = A or B)	0=OFF 1=ON					
ET1BAB	2.5w XMTR-B Arm	0=Safe 1=Arm					
ET1BEB	2.5w XMTR-B Enable	0=Ena 1=Dis					

LINK 2 Communications Subsystem

Variable Name	Description	State Assignments	Critical Low	Nominal Low	Nominal High	Critical High	Nominal
C+150V	: +15 VDC Out Voltage Digital Telemetry Unit		15.49	15.61	15.97	16.09	15.79
C2A+1V	: +15 VDC Out Voltage Link 2A		0.00	0.00	0.38	0.38	0.19
C2A-1V	: -15 VDC Out Voltage Link 2A		-39.50	-39.50	-38.89	-38.89	-39.20
C2A28V	: +28 VDC Out Voltage Link 2A		0.00	0.00	0.71	0.71	0.36
C2ASPB	Ant. Switch Position Monitor	0=A-HI 1=B-HI					
C2B+1V	: +15 VDC Out Voltage Link 2B		14.70	14.93	15.31	15.54	15.12
C2B-1V	: -15 VDC Out Voltage Link 2B		-15.67	-15.31	-14.82	-14.46	-15.07
C2B28V	: +28 VDC Out Voltage Link 2B		27.29	27.72	28.42	28.85	28.07
CC+50V	: +5 VDC Out Voltage Digital Telemetry Unit		4.92	5.00	5.28	5.36	5.14
CC+60V	: +6 VDC Out Voltage Digital Telemetry Unit		6.00	6.10	6.34	6.39	6.22
CC-23V	: -23.5 VDC Out Voltage Digital Telemetry Unit		-22.22	-22.16	-21.86	-21.80	-22.01
CC-60V	: -6 VDC Out Voltage Digital Telemetry Unit		-5.93	-5.87	-5.73	-5.63	-5.80
CCINIT	Internal Temp. Digital Telemetry Uunit		57.27	62.34	88.72	93.94	75.53
CCS1CV	Calibration Voltage 1 (.15 VDC)		0.12	0.12	0.16	0.16	0.14
CCS2CV	Calibration Voltage 2 (2.51 VDC)		2.49	2.49	2.53	2.59	2.51
CCS3CV	Calibration Voltage 3 (5.01 VDC)		4.85	4.95	5.04	5.04	5.00
EDABRB	Digital Telemetry Unit A Bit Rate	0=1K 1=128K					
EDBBRB	Digital Telemetry Unit B Bit Rate	0=1K 1=128K					
EDTUAB	Digital Telemetry Unit A On/Off status	0=ON 1=OFF					
EDTUBB	Digital Telemetry Unit B On/Off status	0=ON 1=OFF					
EDTUDB	Digital Telemetry Unit A/B Data Status	0=BYPS 1=ENCR					
EKG2AB	2 KGX-28-A Status	0=ON 1=OFF					
EKG2BB	2 KGX-28-B Status	0=ON 1=OFF					
ERBUDB	Redundant Baseband Assembly Unit Data Status	0=BYPS 1=ENCR					
ET2AAB	XMTR-A ARM	0=Safe 1=Arm					
ET2AEB	XMTR-A Enable	0=Ena 1=Dis					
ET2AOB	XMTR-A Status	0=OFF 1=ON					
ET2BAB	XMTR-B ARM	0=Safe 1=Arm					
ET2BEB	XMTR-B Enable	0=Dis 1=Ena					
ET2BOB	XMTR-B Status	0=OFF 1=ON					

Propulsion Subsystem

Variable Name	Description	State Assignments	Critical Low	Nominal Low	Nominal High	Critical High	Nominal
ACJEAB	Attitude Control System Control Jets Enable A -X	0=ENA 1=DIS					
ACJEBB	Attitude Control System Control Jets Enable B +X	0=ENA 1=DIS					
ACTEAB	Attitude Control System Control Thrusters Enable A -X	0=ENA 1=DIS					
ACTEBB	Attitude Control System Control Thrusters Enable B +X	0=ENA 1=DIS					
AGGAEB	Attitude Control System Gas Generator Control A -X	0=ENA 1=DIS					
AGGBEB	Attitude Control System Gas Generator Control B +X	0=ENA 1=DIS					
ASJEAB	Attitude Control System Spin Jets Enable A -X	0=ENA 1=DIS					
ASJEBB	Attitude Control System Spin Jets Enable B +X	0=ENA 1=DIS					
AVVEAB	Attitude Control System Delta V Thrusters Enable A -X	0=ENA 1=DIS					
AVVEBB	Attitude Control System Delta V Thrusters Enable B +X	0=ENA 1=DIS					
GGPRES	Gas Generator Pressure		0.00	0.00	0.00	0.00	0.00
LVLSEL	Level Select for Plenum Pressure		0.00	0.00	0.00	0.00	0.00
P+XFBT	Propellant +X Filter Body Temp.		41.92	44.96	121.90	195.13	83.43
P+XPTP	Propellant Tank Pressure +X Tank		165.60	168.00	175.2	177.6	168.00
P+XTOT	Propellant Tank Temp. +X Tank Outboard		42.26	47.03	63.91	68.48	55.47
P+XVBT	Propellant +X Thruster Bank ISO Valve Temp.		41.92	44.96	121.90	195.13	83.43
PFDIVT	Propellant Temp. Fill / Drain ISO-Valve +Y,-X		35.00	36.00	140.00	156.00	88.00
PGGFPT	Gas Generator Temp.		65.39	70.09	103.82	108.35	86.96
PHLT+T	Propellant Temp. High Level Thruster +X		89.43	94.06	127.16	132.13	110.61
PHLT-T	Propellant Temp. High Level Thruster -X		90.04	95.41	123.78	128.35	109.60
PHT+HT	Propellant Temp. Hydrazine Line High Level Thruster +X		57.61	62.48	93.39	98.84	77.94
PHT-HT	Propellant Temp. Hydrazine Line High Level Thruster -X		57.61	62.48	93.39	98.84	77.94
PPLN1P	Plenum Pressure 1		35.10	50.02	78.63	79.84	64.33
PPLN2P	Plenum Pressure 2		35.12	50.29	78.62	79.42	64.46
PVT+HT	Propellant Temp. Hydrazine Line Delta V Thruster +X		35.00	80.00	139.00	177.00	109.50
PVT-HT	Propellant Temp. Hydrazine Line Delta V Thruster -X		35.00	80.00	139.00	177.00	109.50
PVVT+T	Propellant Temp. Delta V Thruster +X		95.41	100.23	145.84	177.73	123.04
PVVT-T	Propellant Temp. Delta V Thruster -X		90.09	95.41	145.84	177.73	120.63
P-XCLT	Propellant Crossover Line Temp.		42.23	47.10	72.24	77.66	59.67
P-XFBT	Propellant -X Filter Body Temp.		41.92	44.96	121.90	195.13	83.43
P-XPTP	Propellant Tank Pressure -X Tank		165.60	168.00	175.20	177.60	171.60
P-XTIT	Propellant Tank Temp. -X Tank Inboard		40.00	40.00	75.00	121.00	57.50
P-XTOT	Propellant Tank Temp. -X Tank Outboard		42.26	47.03	64.89	69.55	55.96
P-XVBT	Propellant Thruster ISO Valve Temp.		41.92	44.96	121.90	195.13	83.43

Electrical Power Subsystem

Variable Name	Description	State Assignments	Critical Low	Nominal Low	Nominal High	Critical High	Nominal
EAACMB	Battery A Auto Charge Mode	0=MOD2 1=MOD1					_
EADBYB	Battery A Diode Bypass	0=IN 1=BYPS					_
EASBPB	Array Switching Unit Base Panels, prime	0=IN 1=BYPS					_
EASBRB	Array Switching Unit Base Panels, redundant	0=DISC 1=CONN					_
EBACMB	Battery B Auto Charge Mode	0=MOD2 1=MOD1					_
EBAHEB	Battery A Heater Power Enable	0=DIS 1=ENA					_
EBAK1B	Battery A Relay K1 Position Full/Trkl	0=FULL 1=TRKL					_
EBAK2B	Battery A Relay K2 Position Disconnect	0=CONN 1=DISC					_
EBAK3B	Battery A Relay K3 Position Charge Control	0=MANU 1=AUTO					_
EBARDB	Battery A Recondition	0=RCDN 1=OPEN					_
EBARTB	Battery A Recondition Termination	0=MANU 1=AUTO					_
EBBHEB	Battery B Heater Power Enable	0=DIS 1=ENA					_
EBBK1B	Battery B Relay K1 Position	0=FULL 1=TRKL					_
EBBK2B	Battery B Relay K2 Position	0=CONN 1=DISC					_
EBBK3B	Battery B Relay K3 Position	0=MANU 1=AUTO					_
EBBRDB	Battery B Recondition	0=RCDN 1=OPEN					_
EBBRTB	Battery B Recondition Termination	0=MANU 1=AUTO					_
EBCHEB	Battery C Heater Power Enable	0=DIS 1=ENA					_
EBCK1B	Battery C Relay K1 Position	0=FULL 1=TRKL					_
EBCK2B	Battery C Relay K2 Position	0=CONN 1=DISC					_
EBCK3B	Battery C Relay K3 Position	0=MANU 1=AUTO					_
EBCRDB	Battery C Recondition	0=RCDN 1=OPEN					_
EBCRTB	Battery C Recondition Termination	0=MANU 1=AUTO					_
EBDBYB	Battery B Diode Bypass	0=IN 1=BYPS					_
EBVLSB	Bus Voltage Limiter Status	0=HIGH 1=LOW					_
ECACMB	Battery C Auto Charge Mode	0=MOD2 1=MOD1					_
ECDBYB	Battery C Diode Bypass	0=DIS 1=ENA					_
ECUSBI	Shunt Bus Current		0.00	0.00	17.58	18.00	8.79
ECUSBV	Shunt Bus Voltage		9.01	10.04	19.76	20.16	14.90
EED1AV	: +10 VDC Output Voltage Electrical Distribution Unit Converter A		9.94	9.94	10.54	10.54	10.24
EDUVSB	Undervoltage Status	0=OVRD 1=NORM					_
EED1BV	: +10 VDC Output Voltage Electrical Distribution Unit Converter B		9.94	9.94	10.54	10.54	10.24
EED5AV	: +5 VDC Output Voltage Electrical Distribution Unit Converter A		5.17	5.17	5.31	5.31	5.24
EED5BV	: +5 VDC Output Voltage Electrical Distribution Unit Converter B		5.17	5.17	5.31	5.31	5.24
EPBA1T	Battery A Temp. No.1		46.31	49.55	70.64	75.75	60.10
EPBA2T	Battery A Temp. No.2		46.31	49.55	70.64	75.75	60.10
EPBB1T	Battery B Temp. No.1		46.31	49.55	70.64	75.75	60.10
EPBB2T	Battery B Temp. No.2		46.31	49.55	70.64	75.75	60.10
EPBC1T	Battery C Temp. No.1		46.31	49.55	70.64	75.75	60.10

Variable Name	Description	State Assignments	Critical Low	Nominal Low	Nominal High	Critical High	Nominal
EPBC2T	Battery C Temp. No.2		46.31	49.55	70.64	75.75	60.10
EPBSAI	Primary Bus Solar Array Current		12.48	14.46	21.83	23.11	18.15
EPLP2T	Solar Array Panel Temp. 2 -x/+y		-19.11	-19.11	188.97	209.84	84.93
EPSBAI	Battery A Current		-10.00	-0.67	0.50	8.00	-0.09
EPSBAV	Battery A Voltage		29.67	30.09	31.90	32.18	31.00
EPSBBI	Battery B Current		-10.00	-0.67	0.50	8.00	-0.09
EPSBBV	Battery B Voltage		29.67	30.09	31.90	32.18	31.00
EPSBCI	Battery C Current		-10.00	-0.67	0.40	8.00	-0.14
EPSBCV	Battery C Voltage		29.95	30.09	31.90	32.18	31.00
EPSDBV	Shunt Drive Bus Voltage		0.00	0.00	6.04	6.51	3.02
EPSEBT	Shunt Element B Temp. -x		22.23	22.32	120.65	126.00	71.49
EPSEDT	Shunt Element D Temp. +x		22.23	22.32	115.89	120.65	69.11
EPSEET	Shunt Element E Temp.		22.24	22.24	124.88	129.57	73.56
EPSLBI	Sensor Load Bus Current		8.02	8.51	10.59	12.79	9.55
EPSPBV	Primary Bus Voltage		29.89	30.17	31.84	32.11	31.01
EPUC1T	Solar Array Upper Conic Temp. 1 +x/-y		-19.11	-19.11	159.36	209.84	70.13
EPUC2T	Solar Array Upper Conic Temp. 2 +x/-y		-19.11	-19.11	159.36	209.84	70.13
EPUN1T	Solar Array Upper Narrow Cyl. Temp. 1 +x/-y		-19.11	-19.11	80.00	209.84	30.45
ESCLBI	Spacecraft Load Bus Current		4.30	4.54	9.80	10.04	7.17
ESP1PB	Array Switching Unit Paddle Incriment-1, prime	0=DISC 1=CONN					_
ESP1RB	Array Switching Unit Paddle Incriment-1, redundant	0=DISC 1=CONN					_
ESP2PB	Array Switching Unit Paddle Incriment-2, prime	0=DISC 1=CONN					_
ESP2RB	Array Switching Unit Paddle Incriment-2, redundant	0=DISC 1=CONN					_
ESP31T	Solar Array Temp.		-19.11	-19.11	164.08	209.84	72.49
ESP32T	Solar Array Temp.		-19.11	-19.11	164.08	209.84	72.49
ESP3PB	Array Switching Unit Paddle Incriment-3, prime	0=DISC 1=CONN					_
ESP3PB	Array Switching Unit Paddle Incriment-3, redundant	0=DISC 1=CONN					_
ESP4PB	Array Switching Unit Paddle Incriment-4, prime	0=DISC 1=CONN					_
ESP4RB	Array Switching Unit Paddle Incriment-4, redundant	0=DISC 1=CONN					_
SLOART	Struct Temp. Lower Solar Array		-86.57	-86.57	158.76	195.16	36.10
SPXCPT	Structural Temp. Comp. Platform		-9.91	29.62	84.09	102.05	56.86

Appendix 4 – System Manual

Satellite Operator Console Test Bed: System Architecture

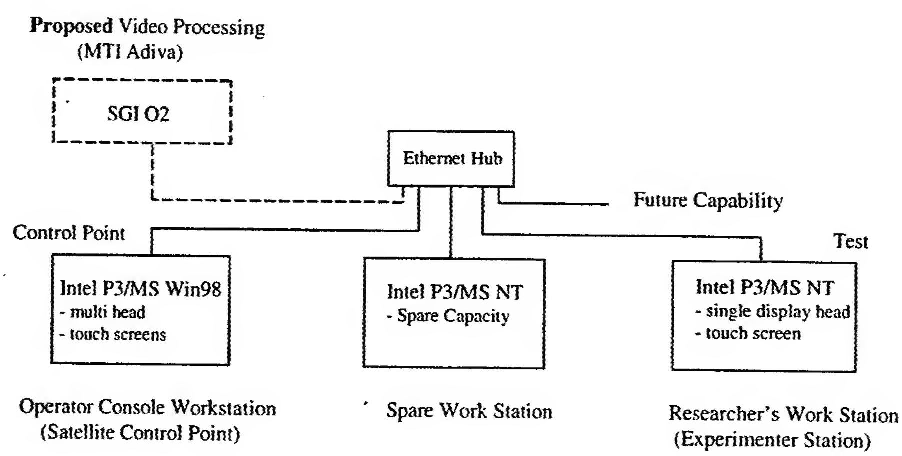
Table Of Contents

Satellite Operator Console Test Bed: Software Architecture

Introduction

This document will describe the software, and the environment in which it executes, for the Space Operations SBIR. Subsequent references to the software will reference specific tasks, or will refer to SpaceOps software when referring to the entire body of code. The system architecture is shown below.

Test Bed System Architecture



The hardware platform is a Dell computer using two varieties of the Microsoft Windows operating system. The Dragon Naturally Speaking SDK was used to integrate voice recognition and text to speech. Dials and gauges were purchased from GMS, in the form of Active X widgets. A graphical editor was used to create displays that show satellite systems. The graphical editor (Altia) can also be used to animate buttons on the displays.

Architecture and Environment

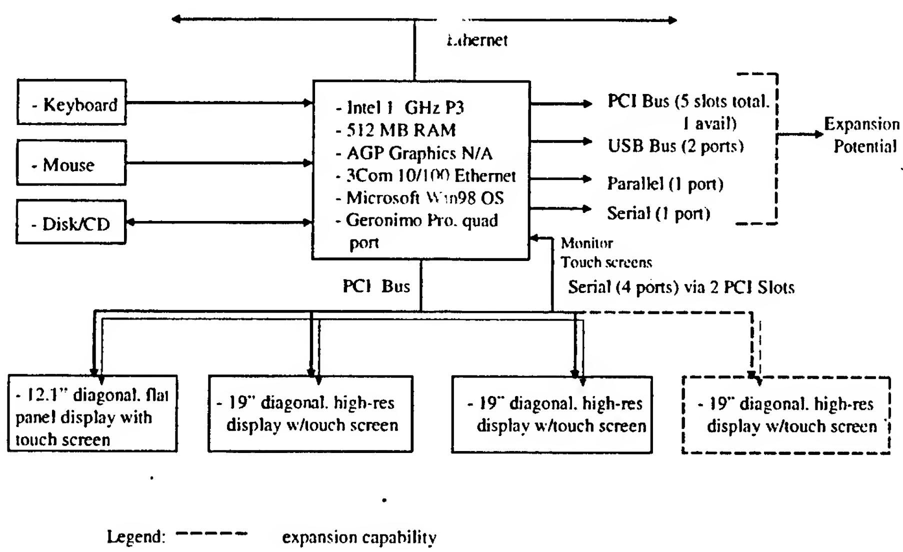
The SpaceOps software is multi tasking, multi-threaded, and runs in a distributed hardware environment. There are three computers used to host Spaceops software. Each computer has a logical name which will be used in the document to reference each machine. The logical name

"ControlPoint" is used for the computer that hosts TimeClient.exe. The logical name "Test" is used for the computer that hosts TimeServer.exe and SpaceOps.exe.

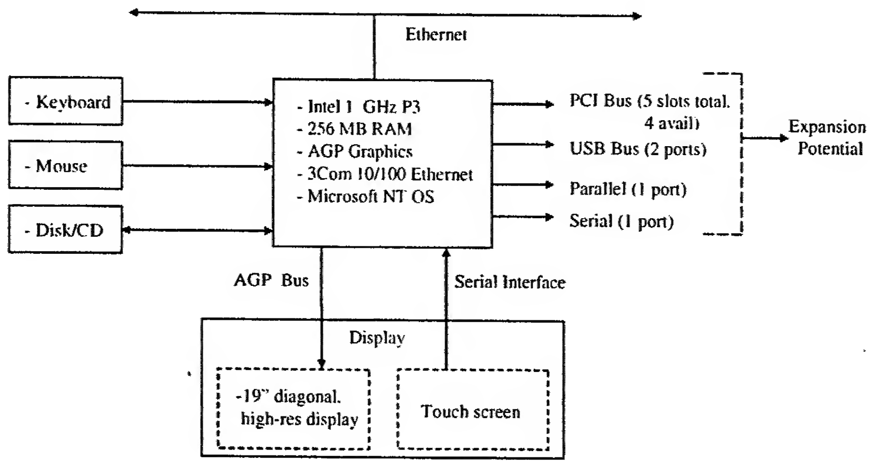
An additional computer is installed in the system. This third computer is not currently used in the simulation, and provides expansion capability for future use. It does not have a logical name at this time.

All inter task communication is done using Ethernet hardware, and TCP/IP software protocol. The hardware architectures of the ControlPoint and Test computers are shown below.

"Control Point" Work Station Architecture

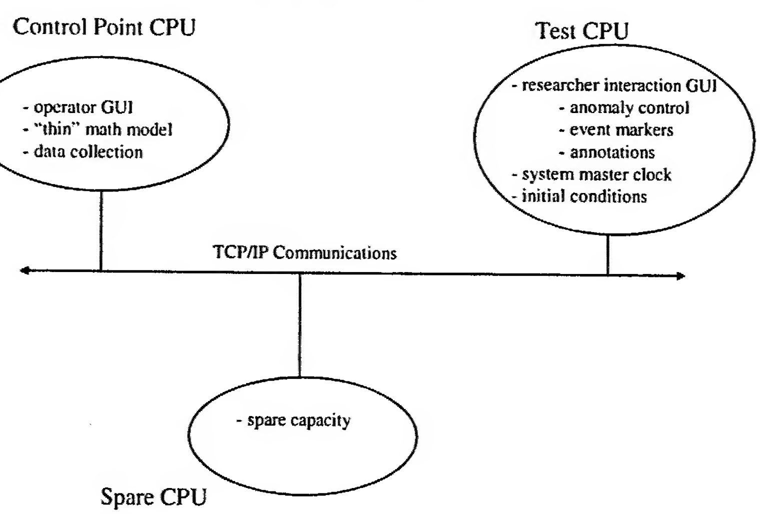


"Test" Workstation Architecture



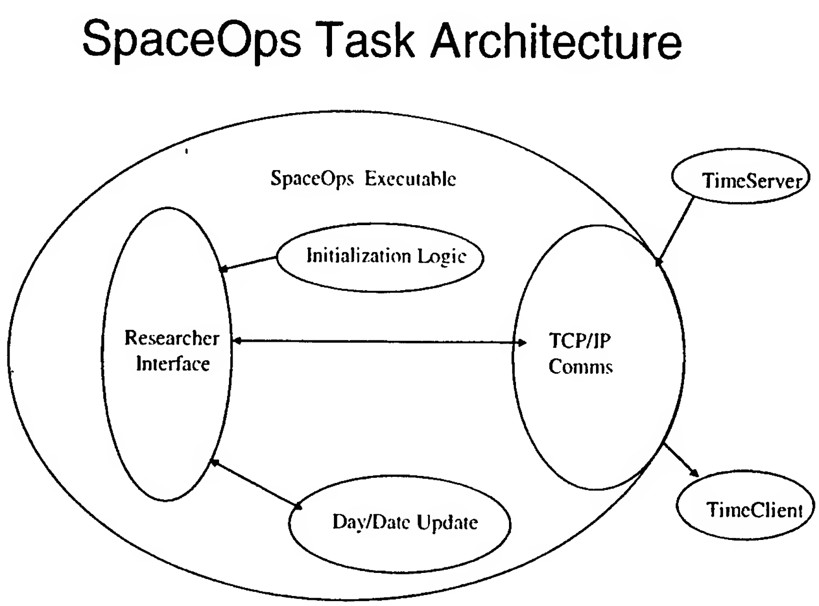
There are three software tasks that comprise the SpaceOps software. These three tasks run on ControlPoint and Test. A third computer is no longer used, and can be considered spare capacity. TimeClient.exe runs on ControlPoint, a Windows 98 machine. TimeServer.exe and SpaceOps.exe run on the Test computer, which is a Windows NT operating system. The software system architecture is shown below.

Overall Software Architecture



SpaceOps.exe

SpaceOps.exe is used as an experimenters graphical User Interface (GUI) for the rest of the SpaceOps software. An experimenter can control the execution and timing of the SpaceOps software from this GUI. The SpaceOps.exe architecture is shown below.



==

MICROSOFT FOUNDATION CLASS LIBRARY DOCUMENTATION: SpaceOps

==

AppWizard created this SpaceOps application

This chapter contains a summary of what you will find in each of the files that make up the SpaceOps application.

SpaceOps.dsp

This file (the project file) contains information at the project level and is used to build a single project or subproject. Other users can share the project (.dsp) file, but they should export the makefiles locally.

SpaceOps.h

This is the main header file for the application. It includes other project specific headers (including Resource.h) and declares the CSpaceOpsApp application class.

SpaceOps.cpp
This is the main application source file that contains the application class CSpaceOpsApp.

SpaceOps.rc
This is a listing of all of the Microsoft Windows resources that the program uses. It includes the icons, bitmaps, and cursors that are stored in the RES subdirectory. This file can be directly edited in Microsoft Visual C++.

SpaceOps.clw
This file contains information used by ClassWizard to edit existing classes or add new classes. ClassWizard also uses this file to store information needed to create and edit message maps and dialog data maps and to create prototype member functions.

res\SpaceOps.ico
This is an icon file, which is used as the application's icon. This icon is included by the main resource file SpaceOps.rc.

res\SpaceOps.rc2
This file contains resources that are not edited by Microsoft Visual C++. You should place all resources not editable by the resource editor in this file.

///

AppWizard creates one dialog class:

SpaceOpsDlg.h and SpaceOpsDlg.cpp - the dialog
These files contain your CSpaceOpsDlg class. This class defines the behavior of your application's main dialog. The dialog's template is in SpaceOps.rc, which can be edited in Microsoft Visual C++.

///

Help Support

hlp\SpaceOps.hpj
This file is the Help Project file used by the Help compiler to create your application's Help file.

hlp*.bmp
These are bitmap files required by the standard Help file topics for Microsoft Foundation Class Library standard commands.

hlp*.rtf
This file contains the standard help topics for standard MFC commands and screen objects.

///
Other standard files:

<u>STDAFX.H AND STDAFX.CPP</u>
These files are used to build a precompiled header (PCH) file named SpaceOps.pch and a precompiled types file named StdAfx.obj.

<u>RESOURCE.H</u>
This is the standard header file, which defines new resource IDs. Microsoft Visual C++ reads and updates this file.

///
Other notes:

AppWizard uses "TODO:" to indicate parts of the source code you should add to or customize.

///

Notes on MTI developed modules and header files:

SocketCode (NetBase.cpp, NetBase.h, NetClient.cpp, NetClient.h, NetQueue.cpp, NetQueue.h, NetStd.h, and SimpleSock.h):
The modules and headers in SocketCode define and implement the Ethernet communications functionality. Socket communications are used to link SpaceOps with the TimeServer and TimeClient executables.

AnomalyTypeDlg.cpp and AnomalyTypeDlg.h;
This module and associated header file defines and implements the anomaly window resource.

cdxCSizingDlg.cpp and cdxCSizingDlg.h:
cdxCSizingDialog.cpp implementation file. (c)1998 Hans Bühler, codex design. Designed to be used with MS VC++ 5.0

This module and associated header file is used to dynamically resize the window components of any window that is being resized by the cursor.

RunTime.cpp and RunTime.h:

The RunTime.cpp module and associated header file is used to update the local times (GMT and ticks past start). The starting date and time are read in via a Win32 API call, and RunTime.cpp is used to update the time. It was also used for data collection purposes early in the development of the project. There is still a lot of legacy code relating to data collection. As of this writing, the data collection is being done in a separate CPU under control of the TimeClient.cpp module.

SpaceOpsDlg.cpp, SpaceOpsDlg.h:

This module and associated header file implements the class behaviors for the SpaceOps GUI. This code implements the main interface between the researcher and the SpaceOps process.

Global.h:

Several global definitions are in this header file.

Anomalies.h:

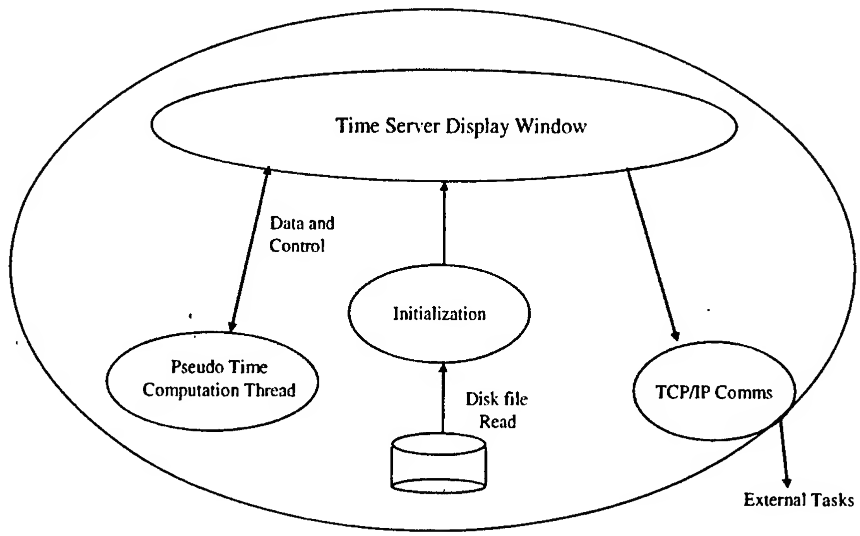
This header file includes a list of all the different satellite variables that can be placed into an anomalous state. This list must correspond (one for one) on the list of selectable anomalies on the SpaceOps GUI. Each anomaly definition value is a unique vector into the anomaly code for a specific variable.

//

TimeServer.exe

TimeServer.exe controls the pseudo time for the application. Given more time, this task would have been folded into SpaceOPs.exe. Due to a major design change in the middle of the SBIR, TimeServer.exe became a stand-alone task, after stripping out much additional functionality that was no longer needed. The TimeServer.exe architecture is shown below.

TimeServer Task Architecture



===

MICROSOFT FOUNDATION CLASS LIBRARY DOCUMENTATION: TimeServer

===

AppWizard created this TimeServer application

This chapter contains a summary of what you will find in each of the files that make up the TimeServer application.

TimeServer.h

This is the main header file for the application. It includes other project specific headers (including Resource.h) and declares the CTimeServerApp application class.

TimeServer.cpp

This is the main application source file that contains the application class CTimeServerApp.

TimeServer.rc

This is a listing of all of the Microsoft Windows resources that the program uses. It includes the icons, bitmaps, and cursors that are stored in the RES subdirectory. This file can be directly edited in Microsoft Visual C++.

res\TimeServer.ico

This is an icon file, which is used as the application's icon. This icon is included by the main resource file TimeServer.rc.

res\TimeServer.rc2

This file contains resources that are not edited by Microsoft Visual C++. You should place all resources not editable by the resource editor in this file.

///

AppWizard creates one dialog class:

TimeServerDlg.h and TimeServerDlg.cpp - the dialog

These files contain your CTimeServerDlg class. This class defines the behavior of your application's main dialog. The dialog's template is in TimeServer.rc, which can be edited in Microsoft Visual C++.

///

Other standard files:

STDAFX.H, STDAFX.CPP

These files are used to build a precompiled header (PCH) file named TimeServer.pch and a precompiled types file named StdAfx.obj

RESOURCE.H

This is the standard header file, which defines new resource IDs. Microsoft Visual C++ reads and updates this file.

///

Other notes:

AppWizard uses "TODO:" to indicate parts of the source code you should add to or customize.

///

Notes on MTI developed modules and header files:

SocketCode (NetBase.cpp, NetBase.h, NetClient.cpp, NetClient.h, NetQueue.cpp, NetQueue.h, NetStd.h, and SimpleSock.h)

The modules and headers in SocketCode define and implement the Ethernet communications functionality. Socket communications are used to link TimeServer with the SpaceOps and TimeClient executables.

RunTime.cpp, RunTime.h

The RunTime.cpp module and associated header file is used to update the pseudo time that is input by an experimenter at run time. The starting date and time are input via a configuration file, and RunTime.cpp is used to update the time

TimeserverDlg.cpp and TimeServerDlg.h

This module and associated header file implements the class behaviors for the TimeServer display window. This code implements the main interface between the researcher and the TimeServer process.

Global.h

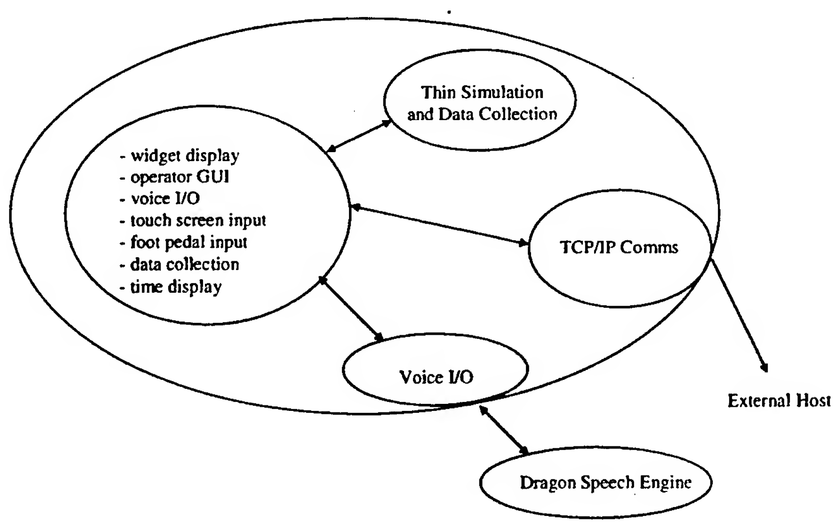
Several global definitions are in this header file.

//

TimeClient.exe

TimeClient.exe is the satellite operators interface into the simulated satellite. All operator I/O is controlled with this task. The I/O includes voice I/O, touchscreen I/O, footpedal I/O and intertask communications. The TimeClient.exe architecture is shown below.

TimeClient Task Architecture



==

MICROSOFT FOUNDATION CLASS LIBRARY DOCUMENTATION: TimeClient

==

AppWizard created this TimeClient application

This file contains a summary of what you will find in each of the files that make up the TimeClient application.

TimeClient.dsp

This file (the project file) contains information at the project level and is used to build a single project or subproject. Other users can share the project (.dsp) file, but they should export the makefiles locally.

TimeClient.h

This is the main header file for the application. It includes other project specific headers (including Resource.h) and declares the CTimeClientApp application class.

TimeClient.cpp

This is the main application source file that contains the application class CTimeClientApp.

TimeClient.rc

This is a listing of all of the Microsoft Windows resources that the program uses. It includes the icons, bitmaps, and cursors that are stored in the RES subdirectory. This file can be directly edited in Microsoft Visual C++.

res\TimeClient.ico

This is an icon file, which is used as the application's icon. This icon is included by the main resource file TimeClient.rc.

res\TimeClient.rc2

This file contains resources that are not edited by Microsoft Visual C++. You should place all resources not editable by the resource editor in this file.

///

AppWizard creates one dialog class:

TimeClientDlg.h, TimeClientDlg.cpp - the dialog

These files contain your CTimeClientDlg class. This class defines the behavior of your application's main dialog. The dialog's template is in TimeCLient.rc, which can be edited in Microsoft Visual C++.

///

Other standard files:

STDAFX.H AND STDAFX.CPP

These files are used to build a precompiled header (PCH) file named TimeClient.pch and a precompiled types file named StdAfx.obj.

RESOURCE.H

This is the standard header file, which defines new resource IDs. Microsoft Visual C++ reads and updates this file.

///

Other notes:

AppWizard uses "TODO:" to indicate parts of the source code you should add to or customize.

///

Notes on MTI developed modules and header files:

SocketCode (NetBase.cpp, NetBase.h, NetClient.cpp, NetClient.h, NetQueue.cpp, NetQueue.h, NetStd.h, and SimpleSock.h)

The modules and headers in SocketCode define and implement the Ethernet communications functionality. Socket communications are used to link TimeClient with the TimeServer and SpaceOps executables.

agauge.cpp and agauge.h

This module and associated header file contains the functions that are used to animate the GMS Active X gauges.

C2A28V.cpp, C2A28Vn, C2A28V.h, and C2A28Vn.h

These modules and associated header file define and implement the gauge (widget) for the C2A28V satellite variable. The gauge displays the value of the variable at a given time. The 'n' modules display normalized values of the variable.

Each gauge (widget) is defined and implemented by it's own cpp module and header file. They are all the same, excepting the name of the module. Each gauge (widget) is an Active X control that must be licensed to run. The TimeClient simulation is capable of displaying roughly 200 different widgets. The naming convention used for each widget was to start with the variable name (c2a28v in the case shown above), and add a lower case 'n' for the normalized values. Some variable names did not allow for this, as in the case of a variable name with a '+' sign. In those cases, the P (plus sign) or M (minus sign) was

capitalized and the rest of the word in lower case e.g. C2B_Minus_1V.cpp. For the sake of brevity, I will not include a written description of each module for the different widgets in this document.

CautionDlg.cpp and CautionDlg.h

This module and associated header file defines and implements the window that displays a list of system caustions. Additional logic regulates when the list is displayed, as opposed to the widget of the variable in a cautions state.

Dgnvoicecmdauto.cpp and dgnvoicecmdauto.h

This module and associated header file define and implement the machine generated IDispatch wrapper class(es) created by Microsoft Visual C++. These wrapper classes allow the MS compiler to interoperate with the Dragon Voice API.

Dgnvoicetxt1.cpp and dgnvoicetxt1.h

This module and associated header file contain the API calls that permit the use of the Dragon speech engine.

Led.cpp and led.h

This module and associated header file define and implement the machine generated IDispatch wrapper class(es) created by Microsoft Visual C++. These wrapper classes allow the MS compiler to interoperate with the GMS led Active X API.

TimeClientDlg.cpp and TimeClientDlg.h

This module and associated header file is the main module for the TimeClient executable. It defines and implements the operator GUI,controls all input and output (voice, USB, Ethernet, serial), and via a subordinate process, it controls the state of the 'thin' simulation. In addition, this module is responsible for data collection and loads in the Dragon voice recognition vocabulary.

Vmenuauto.cpp and vmenuauto.h

This module and associated header file conatins addition Dragon API calls to manipulate the Dragon voice interface.

CautionDlg.cpp and CautionDlg.h

This module and associated header file defines and implements the window that displays a list of system warnings. Additional logic regulates when the list is displayed, as opposed to the widget of the variable in a warning state.

Anomalymap.cpp

This module is an external dependency to the TimeClient project. It must not be 'included' into the project, as this will force a compile of the module. For the sake of clarity and modularity, the logic in

this module was broken out of the TimeClientDlg.cpp source code code, but at compile time, it is re-introduced to the TimeClientDlg.cpp source code, at the appropriate place.

Anomalymap.cpp maps the anomalies (sent from SpaceOps task) to the appropriate logic in the 'thin' simulation.

Datain.h

This header file contains the list of all data collection variables that are currently available. This code is obsolete and has been superseded by logic that collects data on the CONTROL POINT CPU. It was originally used to build a buffer that was sent to SpaceOps.

Displaydialog.cpp

This module is an external dependency to the TimeClient project. It must not be 'included' into the project, as this will force a compile of the module. For the sake of clarity and modularity, the logic in this module was broken out of the TimeClientDlg.cpp source code code, but at compile time, it is re-introduced to the TimeClientDlg.cpp source code, at the appropriate place.

Displaydialog.cpp contains all the widget Display functions for the entire list of variables.

Messagemap.cpp

This module is an external dependency to the TimeClient project. It must not be 'included' into the project, as this will force a compile of the module. For the sake of clarity and modularity, the logic in this module was broken out of the TimeClientDlg.cpp source code code, but at compile time, it is re-introduced to the TimeClientDlg.cpp source code, at the appropriate place.

Messagemap.cpp maps the result of the Dragon speech engine to a specific set of logic that will respond via voice or the display of a widget, when a phrase is recognized.

//

Appendix 5 – Software User's Manual

Table of Contents

Creation of Voice Dialogs

Abstract

One of the key user interface features in the satellite controller test bed is the voice I/O system. This system is based on Dragon *Naturally Speaking*. In order to extend the voice I/O system

This document outlines how to create an new dialog in the Timeclient program for the Satellite Operator Console Test bed project. Once an new dialog is created it can be displayed either using a voice command or when a caution or warning Dialog list is called up and there are less than 6 warnings or cautions whichever the case may be.

There is a standard naming convention for dialogs, classes, variables etc in the Timeclient program which should be strictly adhered to. Each naming convention is given as required throughout the Method.

Method

1. Open the d:\PostBTI\Timeclient workspace from the appropriate folder.
2. Once the workspace is open on the Left hand side select the resources tab and highlight one of the already pre-created widget dialogs. By holding down Ctrl –c keys the dialog will be copied. Press the ctrl- V keys the dialog will be pasted onto the workspace. **Note:** For convenience sake dialogs are copied rather than recreating them from scratch which is very time-consuming.
3. Double left click on the newly created dialog. It should display. Its obviously the same as the copied one.
4. Right click outside the new dialog and select the properties tab and rename it. The convention for naming dialogs is the variable name in uppercase characters with a preceedining IDD_. E.g. Variable name = CCINIT
   Dialog name = IDD_CCINIT –Actual Value dialogs
   Dialog name = IDD_CCINITn – Normalized dialogs
5. Next the GMS widget will have to be altered to suit the variable values. Right click on the widget and select **"GMS Angular Gauge Active X control object" –Properties** . A window should appear
   a. Rename the control using the variable name with a preceeding IDC_
      Eg Variable name = CCINIT
      Dialog name = IDC_CCINIT – Non –Normalized dialogs
      Dialog name = IDC_CCINITn – Normalized dialogs
   b. Change the caption to the variable name. The variable name should end with an 'n' if the dialogs a normalized dialog.

6. The Lower and upper limits must be calculated next. Limits[0] to Limit[3] are already given on the attached sheet. The Lower limit will be called Limit[-1] and the upper Limit will be called Limit[4]. The first limit (Limit[-1]) is calculated as follows Limit[1] - (Limit[2] –Limit[0]). The final limit (Limit[4])is Limit[2] + (Limit[3] –Limit[1]).
7. Select the Scales tab and set the **MIN** and **MAX** limits to the Limit[-1] and Limit[4] values respectively.
8. Next select the Annulars tab. From the list of variable limits attached to this document adjust the annulars as follows.
    a. There should be five different annulars on for each range. Enter limit[-1] into the start value for annular zero. The second given limit, Limit[0] is entered as the stop value. Select Annular ID 1 the start value for this is limit[0] and the stop value is limit[1] for the next annular ID the start value is limit[1] and stop value is limit[2] etc, etc. Continue to fill in the Annular values untill all 5 annulars are complete.
    b. If the dialog is a Normalized dialog calculate the limits as before except calculated them as a % of the Nominal value. Eg. If Limit[0] = 57.27 and Nominal = 75.53 Limit[0] Normalized = (Limit[0]/Nominal)*100 =75.82%. Do this for all the values and enter them as before.

9. Now select the needle tab and position the Needle value slider at the lowest possible limit.
10. Finally select the Tics tab. In the AutoTic tab increment / decrement the TicCount(Major Tic) value until there are enough tics displayed on the widget to show the scale. Close the properties window.
11. The new dialog resource window should still be open. With the new dialog selected press the Ctrl – w keys to invoke class wizard. You should be prompted to select a new class. Enter a class name using the following naming convention. Preceed the variable name with a upper case C.

    E.g. Variable name = CCINIT

    Class Name = CCCINIT

    For Normalized values the class name should finish with a 'n'.

    Click OK.
12. Still in ClassWizard click on IDC_CLOSE in the ObjectsID box. In the Messages box click on BN_CLICKED and press the ADD FUNCTION button which has just highlighted. You will be prompted for a member function name, leave the default and click OK.
13. Still in ClassWizard click on the Member variables tab. There should be 2 control ID's present. Click on the control ID you entered for the widget in line 5a. and press add variable. Enter the variable name preceeded by a lowercase m_. If the widget is for a normalized value place an 'n' at the end of the variable name. Click OK and get out of class wizard by clicking OK.
14. Once back in the workspace click on the file view tab. A new .cpp file will have been created called **Variablename.cpp** Eg.CCINIT.cpp. Open it by double clicking on it. Now by comparing it to a previously created file for a dialog it should be easily altered as follows
    a. Change the constructor function by replacing CWnd with CTimeClientDlg*. Put the following line at the end of the constructor

, m_pParent(pParent)
Eg. **Change**: CCCINIT::CCCINIT(CWnd* pParent /*=NULL*/)
: CDialog(CCCINIT::IDD, pParent)
To
CCCINIT::CCCINIT(CTimeClientDlg* pParent /*=NULL*/)
: CDialog(CCCINIT::IDD, pParent), m_pParent(pParent)

b. Next within the constructor braces place the following line:
Create(Class Name::IDD,pParent); E.g. Create(CCCINIT::IDD,pParent); the class name is the one set in line 12 of these instructions.
This creates the dialog resource.

c. Scroll down to the bottom of the file. There should be a OnClose constructor function Eg. **void CCCINIT::OnClose()**
We are going to create a new function called DeleteDialogVariable Name above the OnClose function.

```
void CCC_Plus_60Vn::DeleteDialogVariableName()
{
        DestroyWindow();
        delete Dlg handle;
        Dlg handle= NULL;
        m_pParent->ScanArray(Voice command);//If present in DialogState array
        //Scan the array and reset the required value to –1
}
```

Note: The voice command variable can be found on the voice commands sheet attached.

d. In the OnClose function insert the following line DeleteDialogVariable Name(); This calls the delete dialog function
e. Next above the newly created DeleteDialog variable add the following line.
extern Class name* Dlg Handle;
Eg.extern CCC_Plus_60Vn* pDlgCC_Plus_60Vn;

If there are difficulties implementing any of this code refer to a previously created dialog .cpp file.

15. Next open the variable name header file **(Variablename.h)**in the header file folder in the file view Eg.CCINIT.h. Open it by double clicking on it.
 a. Include the header file #include "TimeClientDlg.h".
 b. Within the variable class public declaration change the line constructor to accept only a pointer from class CTimeClientDlg.
 E.g. CCCINIT(CWnd* pParent = NULL); goes to.
 CCCINIT(CTimeClientDlg* pParent = NULL);
 c. Next add the declaration of the DeleteDialog function within the public under neat the line **//}}AFX_DATA**
 Eg //}}AFX_DATA
 afx_msg void DeleteDialogCCINIT();
 d. In the protected declaration add the following line:

CTimeClientDlg* m_pParent;

16. Now open TimeClientDlg.cpp. Do the following:
    a. Include the newly created header file below all the previously created header files. e.g. Insert the lines #include "CCINIT.h".
    b. Still in TimeClientDlg.cpp create a pointer to the newly created class underneath all the previously created pointers. I.e. Classname *pDlgVariablename. E.g. CCCINIT* pDlgCCINIT;

17. Press the F7 function key to compile the file. The file should compile without any errors. If it doesn't retrace the steps taken making sure the syntax is correct and all naming conventions have been followed.
18. Still in TimeClientDlg.cpp go to the GenerateData function. Within this function there is a section for each variable where the limits are defined and a value calculated. Scroll down to this section. Add the following lines:

```
if(pDlgHandle)
{
        pDlgHandle ->ControlVariable..SetNeedleValue(Value_variable name);
}
```

The control variable is the one set in line 14 of these instructions.

Eg.
```
if(pDlgCCINIT)
{
        pDlgCCINIT->m_CCINIT.SetNeedleValue(Value_CCINIT);
}
```

19. Open displaydialog.cpp in the external dependencies folder . Scroll down to the appropriate function and insert the following code under **Close_Link2Comm();**

```
if(!pDlgHandle)//If the dialog box in not already displayed
                    //display it.
        {
                pDlgHandle = new ClassName(this);
                AreaDisplay(voice command,&areacalled);
                pDlgHandle ->SetWindowPlacement(&areacalled);
                SetWindowPos(NULL,0,0,0,0,SWP_NOSIZE);
                //Resets the focus to the main window
        }
```

E.g.
```
if(!pDlgCCINIT)//If the dialog box in not already displayed
                                        //display it.
        {
                pDlgCCINIT = new CCCINIT(this);
                AreaDisplay(vs_CCINIT,&areacalled);
                pDlgCCINIT->SetWindowPlacement(&areacalled);
                SetWindowPos(NULL,0,0,0,0,SWP_NOSIZE);
                //Resets the focus to the main window
        }
```

20. Finally open TimeClientDlg.cpp once again and go to the **DeleteDialog** function. This is a large switch statement which contains a case for every dialog. For the Dialog that has

just been created create a new case statement identical to the previous ones except with a different variable name.

E.g.

```
case vs_CC_MINUS_60V:
{
        if(pDlgCC_Minus_60Vn && Normalized == 0)
        {
                pDlgCC_Minus_60Vn->DeleteDialogCC_Minus_60Vn();
        }
        if(pDlgCC_Minus_60V && Normalized == 1)
        {
                pDlgCC_Minus_60V->DeleteDialogCC_Minus_60V();
        }
}
```

21. Press the control key F7 on the keyboard to compile the new dialog. It should compile without any problems. If there are any compile errors double –click on the appropriate on a error and see which line of code the error relates to. If its one of the lines of code that you entered check to make sure it was entered correctly.
22. If the program complies without any errors execute it by pressing the control(Ctrl) key and F5 simultaneously. Once the TimeClient Dialog is displayed turn on the Microphone by clicking on its Icon in the windows taskbar. Speak the command to display the dialog into the microphone and the dialog should display. The list of voice commands is detailed in Commands.h. Once the Dialog is displayed test the following:
    a. Make sure the needle is moving on the dialog widget. If its not recheck the code implemented in section 16.
    b. Click on the close button make sure the dialog disappears.

Editing Widgets

Abstract

The graphics used to show the current values of variables in the satellite controller HSI workstation are created using the Instrumentation Acti\ ..ibrary software (Global Majic Software, Inc., Huntsville, AL). The values of the varia... are generated in the satellite simulation software and used by the widgets to position the pointer or otherwise indicate the current values. This document describes the process for altering the appearance of these widgets.

Method

1. Open *TimeClient.dsw*. This can be done by clicking on the *Shortcut To TimeClient.dsw* icon on the desktop. Alternatively, you can go into *D:\PostBTN\TimeClient* folder and clicking on *TimeClient.dsw*.

This will open a Microsoft Visual Studio window.

2. In Visual Studio, click on the *Resource View* tab in the left hand window.

3. A "tree" should appear in the left hand window. This "tree" shows the directory structure. The root folder is named *Time Client Resources*. The folders inside *Time Client Resources* are:
Dialog
Icon
String Table
Version.

Expand the *Dialog* folder by clicking on it.

4. When the *Dialog* folder is expanded, there will be a large number of entrie These all begin with "IDD_". Scroll through the list until you find the name of the widget you wish to modify. The names of the variables are contained in the "IDD_" files, so you can identify the widget you are interested in easily.

5. Click on the widget you wish to modify. This will cause the widget to appear in the right hand window of Visual Studio.

6. Move the cursor to the widget and right click. This will cause the widget to be "hooked", and a pop up window to appear.

7. Click on the *Properties* option in the pop up window. Clicking on *Properties* will cause another pop up window to appear. This new pop up window has a series of tabs along the top. You can scroll the tabs using the arrows to the right of the tabs.

8. For this example, we are going to set the scale on the strip chart that shows time history data for CPA1AT. We are going to set the display limits from 1.84 to 73.26. There will be five bands within these limits shown in the display. These bands are:

Low Red	from 1.84 to 17.13
Low Yellow	from 17.13 to 22.24
Nominal	from 22.24 to 47.75
Upper Yellow	from 47.75 to 52.86
Upper Red	from 52.86 to 73.26

9. Click on the *Tracks* tab.

10. Enter the maximum allowable data value (73.26) in the field labeled "Max". Enter the minimum allowable data value in the "Min" field

9. Enter the largest value to be displayed in the field labeled "Display Max", and the smallest value to be displayed in the field labeled "Display Min". In most cases, these will be the same as the values in the "Max" and "Min" fields. In no case can the value in "Display Min" be smaller than the value in "Min", nor can the value in "Display Max" be greater than the value in "Max".

10. Click on the *Track Bands* tab. Set the value in the "Track Bands" field to 5, assuming that you want five bands.

11 Set the value in the *Track Band ID* field to 0. (Zero is the lowest band. Band numbers begin with zero rather than 1.) Enter the lower boundary of tack 0 (1.84) into the "Track Band Start" field. Then enter the upper boundary (17.13) into the "Track Band Stop" field.

12. Click on the color button. A palette of colors will appear. Select the desired color for this band. (In our example, the lowest band will be red. Subsequent bands will be yellow, green, yellow, and finally red.)

13. Click "OK".

14. Enter the limits and set the color of the other bands. This is done by repeating steps 10 to 12 for each band. As you make these changes, they will be reflected in the image of the widget.

15. When you have finished configuring the widget, click on the "Close window" button. This button is the "x" in the upper right hand corner of the window.

16. You can now close the window showing the widget itself by clicking on the "Close window" button. This button is the "x" in the upper right hand corner of the window.

17. When you press the exclamation point (or press <ctrl-F5) the program will recompile. When you run the program your changes should take effect. This action saves your changes/

18. The process for altering other widgets, dial gauges, is similar. However, the details of some of the menus differ.

Creation of Pass Plan Screens

Abstract

This section describes the creation of pass plan screens. These screens are created in Designer (Altia, Colorado Springs, CO). The pass plans provide the controller a means to accomplish tasks without requiring memorization of the steps that need to be performed and without requiring reference to external reference materials.

The pass plans contain a series of pre-arranged steps. Performing these steps accomplishes one or more tasks that the controller is attempting to complete. In order to accomplish these tasks, commands may be up-linked to the satellite when the controller executes a step in the pass plan. (Not all steps up-link commands.) The pass plans also instruct the controller to assess the values of measurands of interest and, often, to compare the observed values with desired values. Based on the results of such comparisons, the controller makes a decision regarding subsequent actions to be taken.

Method

In the pass plans developed as part of the HSI, the values of measurands of interest are displayed in lose proximity to the decision cue in the pass plan that requires that value. This spares the

This document describes how to make Pass Plans with the Altia Graphics editor.

DIRECTORY STRUCTURE

The pass plans are created in two main steps, a graphics step and a coding step. The model logic to change state is embedded in TimeClient.exe, while the logic to animate the graphics is embedded in Altiart.exe. The executable Altiart.exe needs to be resident in the D:\PostBTI\bin directory.

The Altia design files (.dsn extension) are located in D:\PostBTI\PassPlan\. The pass plan specific logic for each pass plan is also located here. This logic is used to bridge the gap between TimeClient activity, and the expected Altia result.

Three ASCII text data files are used to send data between TimeClient and Altia. With respect to the pass plan logic, TimeClient_Write is a read only file which contains data that TimeClient has written. The file TimeClient_Binary is used to read the state of the binary variables from TimeClient, and write state changes as a result of operator inputs. The file TimeClientBinary_Anomaly is used to read the state of the anomalies which have been entered

by the experimenter at the DataOps console. These three files must be resident in the same directory as the TimeClient.exe.

BUILDING A NEW PASS PLAN WITH ALTIA

The first step to building a new pass plan is the creation of the graphical interface using Altia.

The graphical pass plans in this experiment were built using Altia Design Release 2.30, so knowledge of Altia will be necessary to create new pass plans. We will use its native terminology to during the following steps. To learn Altia, we recommend taking the tutorials that come with it.

The best way to begin a new pass plan is to start with an old one since this experiment already includes several pass plans. Choosing one that is similar to the pass plan you wish to create will be the best starting point. Try finding one with a similar number of pages or one with pages of similar complexity to your new pass plan.

Make a copy of the old Altia pass plan files (both the .dsn and .rtm files) that you have chosen, and name them as your new pass plan. Then open the new .dsn file in the Altia Graphics Editor. You'll have to zoom in on the upper left corner of the *Main Altia View* at a magnification of 16 to see the objects in these files.

These are the basic objects that you'll find in one of the Altia .dsn files provided with this experiment:

Cursor. The cursor is the arrow icon on the left side of the *Main Altia View*. It is composed of 3 simple shapes that have been *grouped* together.
Pages. Each page is a *group* of smaller objects. Successive pages are aligned at x/y coordinates (0, 950), (0, 900), (0, 850), etc., to make them easy to view using the Altia API. By *focusing in* on a page group, the following items can be accessed:
Endpoints. These are the arrows at the top and bottom of each page. Like the cursor, they are *groups* of simple shapes.
Vertical Line. There is a vertical green line drawn between the endpoints on each page.
Arrows. Small green arrows lie along the vertical line to indicate the direction of flow through the pass plan.
Dots. The color changing icons that lie on the vertical line are used to indicate steps in the pass plan. Like the cursor and endpoints, they are made by *grouping* simple shapes. We will explain how they change color later in this document.
Static Text. This is all the text that never changes. It is sometimes *grouped* with an outline or a black background.
Dynamic Text. This is all the text that can change.

Make New Pass Plan Steps

The first thing to do is to make sure there is a color changing dot for each step in the new pass plan. Individual page groups can be edited by *focusing in* on them. Then the individual page elements can me *moved*, *copied*, *pasted*, or *deleted* to create a new page. Each dot on the pass plan should be accompanied by any static or dynamic text that it needs.

Changing the Cursor Animation

The first and easiest Altia *animation* to change is the cursor animation. In all the existing pass plans, the cursor is manipulated by an *animation* called "cursor." Right-click on the cursor and press the *Animate* button to display the *Altia Animation Editor*. Click on the "cursor" *animation* to select it.

You will need to delete any extra *states* or *states* that you want to change. Cycle the *state* counter in the *Altia Animation Editor*, and delete any such *states*.

To define a new *state*, first cycle the *state* counter to the state you want to define. Next, position the cursor beside the endpoint or color changing dot appropriate to that step in the pass plan. Finally, click the *Define* button in the *Altia Animation Editor*. Now you have a new state in the cursor animation.

Changing a Dot's Color

When proceeding from one step to the next in a pass plan, that step's dot changes from blue to green. The only trick to manipulating this color is being able to select the appropriate shape. Since a dot is actually 3 shapes layered on top of one another, you will need to *focus in* on a dot and select the circle in the middle. Look at the *Altia Animation Editor* now and you will see that the circle you selected has an *animation* called "cursor." It has the same name as the cursor's *animation* so that it will change when the cursor moves through the appropriate steps.

The "Low" and "High" states for the dot should be 0 and some other integer, respectively. To alter when the dot changes color, first cycle the *state* until you come to the *high state* for the dot. Delete this state. Cycle the *state* until the cursor has moved just below the dot. If color changing circle that you have selected isn't already green, set it's color to green. Then click the *Define* button to set this state.

What you should have now is two states defined for the dot. *State* 0 should turn the dot to blue, and this new *state* should turn the dot to green.

Creating Dynamic Text Boxes

It is easier to copy and paste dynamic text boxes to and from existing pass plans, but this is how you would create one from scratch. Text input boxes can be found by clicking on the *Models* button and loading "TEXTIO.DSN" from the MODELS\ directory. These boxes can then be copied and pasted into your .dsn file.

Since we do not what users to be able to input values into these text boxes, we delete all the *stimuli* that can affect them. To do this, select a text input box, and click on the *Stimulate* button. This will show a list of all the *stimuli* that affect the text box. Delete them all.

You will notice two types of dynamic text in the existing pass plans, ones with light blue backgrounds and ones that are plain black text. Since these text input boxes are nothing more than *grouped* shapes, you can *focus in* on them and delete the light blue backgrounds if you wish.

CHANGING DYNAMIC TEXT

There are two ways that you will likely need to change dynamic text in your new Altia interface, either automatically with Altia or from the controlling program using the Altia API. If you open the *Altia Animation Editor* and select one of your dynamic text objects you can see a long list of animations that affect it. The ones of interest to us are the "text" and "integer" *animations.* If you are starting from an existing pass plan, either of these *animations* may have already been renamed to something like "text_3" or "EP1ASB."

To enable a "text" animation to be changed using the Altia API, all you need to do is give it a descriptive name, such as "EP1ASB." The rest of the work is done with the code. If you plan to set a numeric value instead of text, you would want to rename the "integer" *animation.*

The text messages in the existing pass plans that change from "Ready to Load" to "Command Sent" are changed automatically by Altia. Your first instinct might be to change the text and set new cursor *animation* states. However, Altia doesn't allow this. Instead, you must use the *Altia Control Editor.*

Select an automatically changing dynamic text object and click the *Control* button. This will display the *controls* associated with this dynamic text. Notice that these *controls* reference two cursor states just like a color changing dot. The logic is basically the same as for the dots, except that we have to use the *Control Editor* instead of the *Animation Editor.*

INTEGRATING NEW PASS PLANS WITH THE CODE BASE

In order to integrate new pass plans with the existing code base, several steps need to be performed. All steps are critical, but it is not necessary to complete them in any particular order. The ordering of the steps in this document is a suggestion only.

New Voice Command. In messagemap.cpp, add two new cases to the statement, by copying a block of code from a previous pass plan case (such as v_PassPlanOneShow). The two case

statements should reflect the display and removal of a pass plan. At this time, just copy the statements. The next step will detail required changes for the new pass plan. It will be necessary to create the case statement variable as well (v_PassPlanOneShow) in voicedefine.h in order to compile.

In the file commands.h, add the actual voice command syntax statement(s). Rebuild the TimeClient executable to verify that there are no new compile errors as a result of the changes.

<u>Link TimeClient to the Altia Design File</u>. In messagemap.cpp, go to the newly created case statement. In the "CreateProcess" argument list, change all references to the old pass plan path name to the new path name. It will likely be necessary to create any new folders included in the path name. Copy the code from a previous pass plan to the new pass plan folder while you are creating the new pass plan name.

<u>Create a new pass plan state variable in thin.h (e.g. PPlan2Status)</u>. There are several modifications to TimeClientDlg that are necessary. Start by defining ClosePassPlanTwo in TimeClientDlg.h. In TimeClientDlf::DestroyWindow(), add the Close_NewPassPlan() at the end of the list of pass plans to close. In TimeClient::Print_BinaryFile(int n), you must add a new case or develop a new comprehensive strategy to load the TimeClient_Binary file, which is used to pass data to the Altia run time executable. Note that Print_BinaryFile is an overloaded function – be sure to modify the right one. Rebuild the TimeClient executable to verify that there are no new compile errors as a result of the changes.

At this point, TimeClient is fully linked to the Altia executable. You should be able to display an Altia design file, in the pathname you created in the first paragraph in this section (3.2) of this document. The next few steps will describe how to animate and 'kill' the Altia process that drives the animation in the pass plan.

<u>Altia Animation Logic</u>. Create a new C++ 'project' in the pathname you created in 2.2 (in the CreateProcess argument list). This project will contain the code that interfaces to TimeClient and the Altia design file that represents the pass plan. Use the 'console app' paradigm when creating the project. If you are a risk taker, you could also copy a previous pass plan project and edit the project files (.dsp, and .dsw). Be aware however, that Microsoft frowns on this approach. However, even if you create a new .dsp file, it would be wise to start with a copy of an existing PassPlan*.c file (where * is some integer). All of what you need to know about controlling Altia design files can be learned by studying one of these .c files. Compile the new project at this time. Proceed to the next step if no compile errors raise their ugly head.

Using 'Show' and 'Remove' commands, you should now be able to display and remove from the display, the new Altia based pass plan. If this is not the case, it is senseless to continue from this point. You will need to revisit all previous steps to be certain that every thing is in order.

<u>Mapping TimeClient Variables to Altia Pass Plan Variables</u>. In each PassPlan*.c file you will find a function timer1sec(). This function contains a switch statement that acts on the value of a design file's "cursor" *animation*. It also contains a series of "if" statements for updated the contents of some dynamic text boxes in the design file.

The first step is to get all the current information about a running pass plan. These two lines of code are important to start with.

```
AtPollEvent((AtConnectId)data, "cursor", &cursor);
BinaryFile_scan();
```

The first gets the current *state* of the cursor. The second gets the current values of the variables that we are displaying and manipulating in the pass plan.

When the cursor moves from one step to the next, one of two things might happen. If the cursor moves from one page to the next, the AtMoveView() function is called to move to a new page group in the *Main Altia View* like this.

```
AtMoveView((AtConnectId)data, 0, 0, 900);
```

If the cursor movement is intended to simulate a command being sent to the satellite, you might set the value one of the state variables and then call BinaryFile_Write() to send the change back to TimeClient.

The "if" statements are used to update dynamic text in the design file. In section 2.5 we discussed renaming "text" *animations* to something more descriptive, such as a state variable name. These new *animation* names can now be accessed using AtSendText() or AtSendEvent() to update text or integers, respectively. The commands look something like this.

```
AtSendText((AtConnectId)data, "EP1ASB", "OFF");

AtSendEvent((AtConnectId)data, "CPA1AT", (int)(Value_CPA1AT + 0.5f));
```

VII. Using The Data Collection System

The Space Operations simulation will collect all of the variable defined in the file DataCollectionList.cpp, subject to the rules of the C++ language syntax. As of this writing, a copy of DataCollectionList.cpp is shown in Appendix A. If a specific variable is not required to be in the list of collected variables, it can be commented out in DataCollectionList.cpp. If it is commented out, after the TimeClient executable is recompiled, that specific variable will no longer be collected.

Data collection needs to be manually turned on with the experimenters interface (SpaceOps.exe). Default is for data collection to be off. If so desired, turn data collection on, select a data rate. The default data rate is 1Hz. Four other rates are selectable by the operator, .5Hz, .2Hz, 4Hz, and 10Hz. The only restriction on the amount of data collected, is the amount of disk space remaining on the ControlPoint computer. There are roughly 360 variables in the default collection list. If none are commented out, and the data rate is set to the fastest rate, the data collection code will collect at the rate of 51MB/hour.

VIII. User Modifications

Screen Layouts

Previous chapters have dealt with the Altia displays, which consist of pass plans, electrical flow diagrams, and communication system flows. This chapter deals with the rest of the desktop real estate which is used by the Space Operation simulation.

All the displays which are not Altia derived, were created using the MS compiler. To modify the displays, the compiler is required as well. All three executables used with the Space Operations simulation (TimeClient.exe, SpaceOps.exe, and TimeServer.exe) are using desk top real estate, and have displays that can be changed. If the user should so desire it, the following procedure will help.

To change the experimenter station desktop display, activate the MS VC++ compiler by double clicking on the SpaceOps.dsw icon. This will activate the MS compiler, and load the source code. In the middle left of the view, there three or four 'view' tabs that can be used to select the class view, resource view, or file view. If not already selected, select the resource view by clicking with the mouse. Expand the SpaceOps resource folder, and expand the dialog folder that shows up. The desktop displays are shown as files in the expansion. Double clicking on a file will cause the resource to show up in the editing window. To change the look and feel of the desktop displays, use the mouse to select and manipulate the display.

Adding Additional Voice Commands

To add additional commands to the system, you need to modify the header file 'commands.h'. The syntax is complicated, so it is recommended that if you want to add a string to be recognized, it is best to copy an existing phrase, and change the text inside the quotes. The first argument to the m_VMenu.Add Dragon API, is a constant that is unique to the phrase being recognized. It needs to be created and added to 'voicedefine.h. Change the text (inside the double quotes in the second argument) to the required phrase. Having done made these changes, it is also necessary to add an additional case statement in messagemap.cpp. This is best done by copying anexisting case, and making the relevant changes. After the changes are made, it will be necessary to compile and link the TimeClient executable.

Appendix A. Data Collection Variable List

As of delivery date, the list of variables being collected is shown below. This list is part of the source code delivery of the simulation. It is identical to the file DataCollectionList.cpp.

```
// FILE: DataCollectionList.cpp
// Created: 03 Dec 01
// Updated: 04 Dec 01
// Converted to C++: 12/18/01
//
// This file is used to add or subtract to the list of variables
// being collected. Any text added to this file is subject to
// the syntax rules of the 'C' computer language. Changes need
// to be compiled , before they will take effect.
//
// All of the variables in this list will be collected except
// those that are "commented" out. In order to remove a variable
// from data collection simply "comment out" the line of
// Code containing the unwanted variable(s). Use a '//' (double
// slash) to comment out any variable.
//


// ************************************************
// ** Data Collection for Satellite Control Test bed **
// ************************************************


	fprintf(dataFile,"%d,",dcMode);			// start/stop/pause/resume
	fprintf(dataFile,"%d,",dcEventCount);			// operator entered
	fprintf(dataFile,"%d,",dcAnomalyFlag);		// operator entered
	fprintf(dataFile,"%d,",dcAnomalyType);		// operator entered
	fprintf(dataFile,"%s,",dcPseudoTime);			// experiment time
	fprintf(dataFile,"%s,",dcGmt);				// Greenwhich Mean Time
	fprintf(dataFile,"%d,",dcTime);			// wall time in integer format
	fprintf(dataFile,"%d,",dcExpTicks);			// 100 ms ticks past start
	fprintf(dataFile,"%d,",dcCount);			// simulation loop count

// ******************************************************************
// ** Data from the "thin simulation" of the satellite"			**
// **								**
// ** The variable names, values and ranges were provided to MTI by CERES.	**
```

```
// ** The satellite simulated is a DSCS bird.                              **
// **                                                                      **
// ** The variables are grouped by subsystem.  Within each subsystem the   **
// ** variables are listed in groups of five to improve legibility.        **
// ************************************************************************

//
//--------------------------------------------------
//
// ** CONTINUOUS SATELLITE VARIABLES - RAW VALUES **
//
//--------------------------------------------------
//

//
// Link 2 Communications Subsystem - Continuous Variables - Raw Values
//

fprintf(dataFile,"%3.5f,",Value_C_plus_150V);        //C+150V
fprintf(dataFile,"%3.5f,",Value_C2A_plus_1V);        //C2A+1V
fprintf(dataFile,"%3.5f,",Value_C2A_minus_1V);       //C2A-1V
fprintf(dataFile,"%3.5f,",Value_C2A28V);             //C2A28V
fprintf(dataFile,"%3.5f,",Value_C2B_plus_1V);        //C2B+1V
fprintf(dataFile,"%3.5f,",Value_C2B_minus_1V);       //C2B-1V
fprintf(dataFile,"%3.5f,",Value_C2B28V);             //C2B28V
fprintf(dataFile,"%3.5f,",Value_CC_plus_50V);        //CC+150V
fprintf(dataFile,"%3.5f,",Value_CC_plus_60V);        //CC+60V
fprintf(dataFile,"%3.5f,",Value_CC_minus_23V);       //CC-23V
fprintf(dataFile,"%3.5f,",Value_CC_minus_60V);       //CC-60V
fprintf(dataFile,"%3.5f,",Value_CCINIT);             //CCINIT
fprintf(dataFile,"%3.5f,",Value_CCS1CV);             //CCS1CV
fprintf(dataFile,"%3.5f,",Value_CCS2CV);             //CCS2CV
fprintf(dataFile,"%3.5f,",Value_CCS3CV);             //CCS3CV

//
// Link 1 Communications Subsystem - Continuous Variables - Raw Values
//

fprintf(dataFile,"%3.5f,",Value_CCT1AV);             //CCT1AV
fprintf(dataFile,"%3.5f,",Value_CCT1AW);             //CCT1AW
fprintf(dataFile,"%3.5f,",Value_CCT1BV);             //CCT1BV
fprintf(dataFile,"%3.5f,",Value_CCT1BW);             //CCT1BW
fprintf(dataFile,"%3.5f,",Value_CECA5V);             //CECA5V
fprintf(dataFile,"%3.5f,",Value_CECB5V);             //CECB5V
fprintf(dataFile,"%3.5f,",Value_CPA1AT);             //CPA1AT
fprintf(dataFile,"%3.5f,",Value_CPA1AV);             //CPA1AV
```

```
fprintf(dataFile,"%3.5f,",Value_CPA1AW);          //CPA1AW
fprintf(dataFile,"%3.5f,",Value_CPA1BT);          //CPA1BT

fprintf(dataFile,"%3.5f,",Value_CPA1BV);          //CPA1BV
fprintf(dataFile,"%3.5f,",Value_CPA1BW);          //CPA1BW
fprintf(dataFile,"%3.5f,",Value_CT1AMT);          //CT1AMT
fprintf(dataFile,"%3.5f,",Value_CT1BMT);          //CT1BMT

//
// Propulsion Subsystem - Continuous Variables - Raw Values
//

fprintf(dataFile,"%3.5f,",Value_GGPRES);          //GGPRES
fprintf(dataFile,"%3.5f,",Value_LVLSEL);          //LVLSEL
fprintf(dataFile,"%3.5f,",Value_P_plus_XFBT);     //P+XFBT
fprintf(dataFile,"%3.5f,",Value_P_plus_XPTP);     //P+XPTP
fprintf(dataFile,"%3.5f,",Value_P_plus_XTOT);     //P+XTOT
fprintf(dataFile,"%3.5f,",Value_P_plus_XVBT);     //P+XVBT
fprintf(dataFile,"%3.5f,",Value_PFDIVT);          //PFDIVT
fprintf(dataFile,"%3.5f,",Value_PGGFPT);          //PGGFPT
fprintf(dataFile,"%3.5f,",Value_PHLT_plus_T);     //PHLT+T
fprintf(dataFile,"%3.5f,",Value_PHLT_minus_T);    //PHLT-T
fprintf(dataFile,"%3.5f,",Value_PHT_plus_HT);     //PHT+HT
fprintf(dataFile,"%3.5f,",Value_PHT_minus_HT);    //PHT-HT
fprintf(dataFile,"%3.5f,",Value_PPLN1P);          //PPLN1P
fprintf(dataFile,"%3.5f,",Value_PPLN2P);          //PPLN2P
fprintf(dataFile,"%3.5f,",Value_PVT_plus_HT);     //PVT+HT
fprintf(dataFile,"%3.5f,",Value_PVT_minus_HT);    //PVT-HT
fprintf(dataFile,"%3.5f,",Value_PVVT_plus_T);     //PVVT+T
fprintf(dataFile,"%3.5f,",Value_PVVT_minus_T);    //PVVT-T
fprintf(dataFile,"%3.5f,",Value_P_minus_XCLT);    //P-XCLT
fprintf(dataFile,"%3.5f,",Value_P_minus_XFBT);    //P-XFBT
fprintf(dataFile,"%3.5f,",Value_P_minus_XPTP);    //P-XPTP
fprintf(dataFile,"%3.5f,",Value_P_minus_XTIT);    //P-XTIT
fprintf(dataFile,"%3.5f,",Value_P_minus_XTOT);    //P-XTOT
fprintf(dataFile,"%3.5f,",Value_P_minus_XVBT);    //P-XVBT

//
// Electrical Power Subsystem - Continuous Variables - Raw Values
//

fprintf(dataFile,"%3.5f,",Value_ECUSBI);          //ECUSBI
fprintf(dataFile,"%3.5f,",Value_ECUSBV);          //ECUSBV
fprintf(dataFile,"%3.5f,",Value_EED1AV);          //EED1AV
fprintf(dataFile,"%3.5f,",Value_EED1BV);          //EED1BV
```

```
fprintf(dataFile,"%3.5f,",Value_EED5AV);          //EED5AV
fprintf(dataFile,"%3.5f,",Value_EED5BV);          //EED5BV
fprintf(dataFile,"%3.5f,",Value_EPBA1T);          //EPBA1T
fprintf(dataFile,"%3.5f,",Value_EPBA2T);          //EPBA2T
fprintf(dataFile,"%3.5f,",Value_EPBB1T);          //EPPB1T
fprintf(dataFile,"%3.5f,",Value_EPBB2T);          //EPBB2T
fprintf(dataFile,"%3.5f,",Value_EPBC1T);          //EPBC1T
fprintf(dataFile,"%3.5f,",Value_EPBC2T);          //EPBC2T
fprintf(dataFile,"%3.5f,",Value_EPBSAI);          //EPBSAI
fprintf(dataFile,"%3.5f,",Value_EPLP2T);          //EPLP2T
fprintf(dataFile,"%3.5f,",Value_EPSBAI);          //EPSBAI
fprintf(dataFile,"%3.5f,",Value_EPSBAV);          //EPSBAV
fprintf(dataFile,"%3.5f,",Value_EPSBBI);          //EPSBBI
fprintf(dataFile,"%3.5f,",Value_EPSBBV);          //EPSBBV
fprintf(dataFile,"%3.5f,",Value_EPSBCI);          //EPSBCI
fprintf(dataFile,"%3.5f,",Value_EPSBCV);          //EPSBCV
fprintf(dataFile,"%3.5f,",Value_EPSDBV);          //EPSDBV
fprintf(dataFile,"%3.5f,",Value_EPSEBT);          //EPSEBT
fprintf(dataFile,"%3.5f,",Value_EPSEDT);          //EPSEDT
fprintf(dataFile,"%3.5f,",Value_EPSEET);          //EPSEET
fprintf(dataFile,"%3.5f,",Value_EPSLBI);          //EPSLBI
fprintf(dataFile,"%3.5f,",Value_EPSPBV);          //EPSPBV
fprintf(dataFile,"%3.5f,",Value_EPUC1T);          //EPUC1T
fprintf(dataFile,"%3.5f,",Value_EPUC2T);          //EPUC2T
fprintf(dataFile,"%3.5f,",Value_EPUN1T);          //EPUN1T
fprintf(dataFile,"%3.5f,",Value_ESCLBI);          //ESCLBI
fprintf(dataFile,"%3.5f,",Value_ESP31T);          //ESP31T
fprintf(dataFile,"%3.5f,",Value_ESP32T);          //ESP32T
fprintf(dataFile,"%3.5f,",Value_SLOART);          //SLOART
fprintf(dataFile,"%3.5f,",Value_SPXCPT);          //SPXCPT

//
//---------------------------------------------------
//
// ** CONTINUOUS SATELLITE VARIABLES - NORMALIZED VALUES **
//
//-------------------------------------------------------
//

//
// Link 2 Communications Subsystem - Continuous Variables - Normalized Values
//

fprintf(dataFile,"%3.5f,",Norm_C_plus_150V);      //C+150V
fprintf(dataFile,"%3.5f,",Norm_C2A_plus_1V);      //C2A+1V
fprintf(dataFile,"%3.5f,",Norm_C2A_minus_1V);     //C2A-1V
```

```
fprintf(dataFile,"%3.5f,",Norm_C2A28V);             //C2A28V
fprintf(dataFile,"%3.5f,",Norm_C2B_plus_1V);        //C2B+1V
fprintf(dataFile,"%3.5f,",Norm_C2B_minus_1V);       //C2B-1V
fprintf(dataFile,"%3.5f,",Norm_C2B28V);             //C2B28V
fprintf(dataFile,"%3.5f,",Norm_CC_plus_50V);        //CC+150V
fprintf(dataFile,"%3.5f,",Norm_CC_plus_60V);        //CC+60V
fprintf(dataFile,"%3.5f,",Norm_CC_minus_23V);       //CC-23V
fprintf(dataFile,"%3.5f,",Norm_CC_minus_60V);       //CC-60V
fprintf(dataFile,"%3.5f,",Norm_CCINIT);             //CCINIT
fprintf(dataFile,"%3.5f,",Norm_CCS1CV);             //CCS1CV
fprintf(dataFile,"%3.5f,",Norm_CCS2CV);             //CCS2CV
fprintf(dataFile,"%3.5f,",Norm_CCS3CV);             //CCS3CV

//
// Link 1 Communications Subsystem - Continuous Variables - Normalized Values
//

fprintf(dataFile,"%3.5f,",Norm_CCT1AV);             //CCT1AV
fprintf(dataFile,"%3.5f,",Norm_CCT1AW);             //CCT1AW
fprintf(dataFile,"%3.5f,",Norm_CCT1BV);             //CCT1BV
fprintf(dataFile,"%3.5f,",Norm_CCT1BW);             //CCT1BW
fprintf(dataFile,"%3.5f,",Norm_CECA5V);             //CECA5V
fprintf(dataFile,"%3.5f,",Norm_CECB5V);             //CECB5V
fprintf(dataFile,"%3.5f,",Norm_CPA1AT);             //CPA1AT
fprintf(dataFile,"%3.5f,",Norm_CPA1AV);             //CPA1AV
fprintf(dataFile,"%3.5f,",Norm_CPA1AW);             //CPA1AW
fprintf(dataFile,"%3.5f,",Norm_CPA1BT);             //CPA1BT
fprintf(dataFile,"%3.5f,",Norm_CPA1BV);             //CPA1BV
fprintf(dataFile,"%3.5f,",Norm_CPA1BW);             //CPA1BW
fprintf(dataFile,"%3.5f,",Norm_CT1AMT);             //CT1AMT
fprintf(dataFile,"%3.5f,",Norm_CT1BMT);             //CT1BMT

//
// Propulsion Subsystem - Continuous Variables - Raw Values
//

fprintf(dataFile,"%3.5f,",Norm_GGPRES);             //GGPRES
fprintf(dataFile,"%3.5f,",Norm_LVLSEL);             //LVLSEL
fprintf(dataFile,"%3.5f,",Norm_P_plus_XFBT);        //P+XFBT
fprintf(dataFile,"%3.5f,",Norm_P_plus_XPTP);        //P+XPTP
fprintf(dataFile,"%3.5f,",Norm_P_plus_XTOT);        //P+XTOT
fprintf(dataFile,"%3.5f,",Norm_P_plus_XVBT);        //P+XVBT
fprintf(dataFile,"%3.5f,",Norm_PFDIVT);             //PFDIVT
fprintf(dataFile,"%3.5f,",Norm_PGGFPT);             //PGGFPT
fprintf(dataFile,"%3.5f,",Norm_PHLT_plus_T);        //PHLT+T
fprintf(dataFile,"%3.5f,",Norm_PHLT_minus_T);       //PHLT-T
```

```
fprintf(dataFile,"%3.5f,",Norm_PHT_plus_HT);        //PHT+HT
fprintf(dataFile,"%3.5f,",Norm_PHT_minus_HT);       //PHT-HT
fprintf(dataFile,"%3.5f,",Norm_PPLN1P);             //PPLN1P
fprintf(dataFile,"%3.5f,",Norm_PPLN2P);             //PPLN2P
fprintf(dataFile,"%3.5f,",Norm_PVT_plus_HT);        //PVT+HT
fprintf(dataFile,"%3.5f,",Norm_PVT_minus_HT);       //PVT-HT
fprintf(dataFile,"%3.5f,",Norm_PVVT_plus_T);        //PVVT+T
fprintf(dataFile,"%3.5f,",Norm_PVVT_minus_T);       //PVVT-T
fprintf(dataFile,"%3.5f,",Norm_P_minus_XCLT);       //P-XCLT
fprintf(dataFile,"%3.5f,",Norm_P_minus_XFBT);       //P-XFBT
fprintf(dataFile,"%3.5f,",Norm_P_minus_XPTP);       //P-XPTP
fprintf(dataFile,"%3.5f,",Norm_P_minus_XTIT);       //P-XTIT
fprintf(dataFile,"%3.5f,",Norm_P_minus_XTOT);       //P-XTOT
fprintf(dataFile,"%3.5f,",Norm_P_minus_XVBT);       //P-XVBT

//
// Electrical Power Subsystem - Continuous Variables - Normalized Values
//

fprintf(dataFile,"%3.5f,",Norm_ECUSBI);             //ECUSBI
fprintf(dataFile,"%3.5f,",Norm_ECUSBV);             //ECUSBV
fprintf(dataFile,"%3.5f,",Norm_EED1AV);             //EED1AV
fprintf(dataFile,"%3.5f,",Norm_EED1BV);             //EED1BV
fprintf(dataFile,"%3.5f,",Norm_EED5AV);             //EED5AV
fprintf(dataFile,"%3.5f,",Norm_EED5BV);             //EED5BV
fprintf(dataFile,"%3.5f,",Norm_EPBA1T);             //EPBA1T
fprintf(dataFile,"%3.5f,",Norm_EPBA2T);             //EPBA2T
fprintf(dataFile,"%3.5f,",Norm_EPBB1T);             //EPPB1T
fprintf(dataFile,"%3.5f,",Norm_EPBB2T);             //EPBB2T
fprintf(dataFile,"%3.5f,",Norm_EPBC1T);             //EPBC1T
fprintf(dataFile,"%3.5f,",Norm_EPBC2T);             //EPBC2T
fprintf(dataFile,"%3.5f,",Norm_EPBSAI);             //EPBSAI
fprintf(dataFile,"%3.5f,",Norm_EPLP2T);             //EPLP2T
fprintf(dataFile,"%3.5f,",Norm_EPSBAI);             //EPSBAI
fprintf(dataFile,"%3.5f,",Norm_EPSBAV);             //EPSBAV
fprintf(dataFile,"%3.5f,",Norm_EPSBBI);             //EPSBBI
fprintf(dataFile,"%3.5f,",Norm_EPSBBV);             //EPSBBV
fprintf(dataFile,"%3.5f,",Norm_EPSBCI);             //EPSBCI
fprintf(dataFile,"%3.5f,",Norm_EPSBCV);             //EPSBCV
fprintf(dataFile,"%3.5f,",Norm_EPSDBV);             //EPSDBV
fprintf(dataFile,"%3.5f,",Norm_EPSEBT);             //EPSEBT
fprintf(dataFile,"%3.5f,",Norm_EPSEDT);             //EPSEDT
fprintf(dataFile,"%3.5f,",Norm_EPSEET);             //EPSEET
fprintf(dataFile,"%3.5f,",Norm_EPSLBI);             //EPSLBI
fprintf(dataFile,"%3.5f,",Norm_EPSPBV);             //EPSPBV
fprintf(dataFile,"%3.5f,",Norm_EPUC1T);             //EPUC1T
```

```
fprintf(dataFile,"%3.5f,",Norm_EPUC2T);          //EPUC2T
fprintf(dataFile,"%3.5f,",Norm_EPUN1T);          //EPUN1T
fprintf(dataFile,"%3.5f,",Norm_ESCLBI);          //ESCLBI
fprintf(dataFile,"%3.5f,",Norm_ESP31T);          //ESP31T
fprintf(dataFile,"%3.5f,",Norm_ESP32T);          //ESP32T
fprintf(dataFile,"%3.5f,",Norm_SLOART);          //SLOART
fprintf(dataFile,"%3.5f,",Norm_SPXCPT);          //SPXCPT

//
//--------------------------------------------------
//
// ** FLAGS ON CONTINUOUS SATELLITE VARIABLES **
//
//--------------------------------------------------
//

// Link 2 Communications Subsystem - FLAGS
fprintf(dataFile,"%d,",Flag_C_plus_150V);        //C+150V
fprintf(dataFile,"%d,",Flag_C2A_plus_1V);        //C2A+1V
fprintf(dataFile,"%d,",Flag_C2A_minus_1V);       //C2A-1V
fprintf(dataFile,"%d,",Flag_C2A28V);             //C2A28V
fprintf(dataFile,"%d,",Flag_C2B_plus_1V);        //C2B+1V
fprintf(dataFile,"%d,",Flag_C2B_minus_1V);       //C2B-1V
fprintf(dataFile,"%d,",Flag_C2B28V);             //C2B28V
fprintf(dataFile,"%3.3f\t",Flag_CC_plus_50V);    //CC+50V
fprintf(dataFile,"%3.3f\t",Flag_CC_plus_60V);    //CC+60V
fprintf(dataFile,"%3.3f\t",Flag_CC_minus_23V);   //CC-23V
fprintf(dataFile,"%d,",Flag_CC_minus_60V);       //CC-60V
fprintf(dataFile,"%d,",Flag_CCINIT);                  //CCINIT
fprintf(dataFile,"%d,",Flag_CCS1CV);             //CCS1CV
fprintf(dataFile,"%d,",Flag_CCS2CV);             //CCS2CV
fprintf(dataFile,"%d,",Flag_CCS3CV);             //CCS3CV

//
// Link 1 Communications Subsystem - FLAGS
//

fprintf(dataFile,"%d,",Flag_CCT1AV);             //CCT1AV
fprintf(dataFile,"%d,",Flag_CCT1AW);             //CCT1AW
fprintf(dataFile,"%d,",Flag_CCT1BV);             //CCT1BV
fprintf(dataFile,"%d,",Flag_CCT1BW);             //CCT1BW
fprintf(dataFile,"%d,",Flag_CECA5V);             //CECA5V
fprintf(dataFile,"%d,",Flag_CECB5V);             //CECB5V
fprintf(dataFile,"%d,",Flag_CPA1AT);             //CPA1AT
fprintf(dataFile,"%d,",Flag_CPA1AV);             //CPA1AV
fprintf(dataFile,"%d,",Flag_CPA1AW);             //CPA1AW
```

```
fprintf(dataFile,"%d,",Flag_CPA1BT);            //CPA1BT
fprintf(dataFile,"%d,",Flag_CPA1BV);            //CPA1BV
fprintf(dataFile,"%d,",Flag_CPA1BW);            //CPA1BW
fprintf(dataFile,"%d,",Flag_CT1AMT);            //CT1AMT
fprintf(dataFile,"%d,",Flag_CT1BMT);            //CT1BMT

//
// Propulsion Subsystem - FLAGS
//

fprintf(dataFile,"%d,",Flag_GGPRES);            //GGPRES
fprintf(dataFile,"%d,",Flag_LVLSEL);            //LVLSEL
fprintf(dataFile,"%d,",Flag_P_plus_XFBT);       //P+XFBT
fprintf(dataFile,"%d,",Flag_P_plus_XPTP);       //P+XPTP
fprintf(dataFile,"%d,",Flag_P_plus_XTOT);       //P+XTOT
fprintf(dataFile,"%d,",Flag_P_plus_XVBT);       //P+XVBT
fprintf(dataFile,"%d,",Flag_PFDIVT);            //PFDVIT
fprintf(dataFile,"%d,",Flag_PGGFPT);            //PGGFPT
fprintf(dataFile,"%d,",Flag_PHLT_plus_T);       //PHLT+T
fprintf(dataFile,"%d,",Flag_PHLT_minus_T);      //PHLT-T
fprintf(dataFile,"%d,",Flag_PHT_plus_HT);       //PHT+HT
fprintf(dataFile,"%d,",Flag_PHT_minus_HT);      //PHT-HT
fprintf(dataFile,"%d,",Flag_PPLN1P);            //PPLN1P
fprintf(dataFile,"%d,",Flag_PPLN2P);            //PPLN2P
fprintf(dataFile,"%d,",Flag_PVT_plus_HT);       //PVT+HT
fprintf(dataFile,"%d,",Flag_PVT_minus_HT);      //PVT-HT
fprintf(dataFile,"%d,",Flag_PVVT_plus_T);       //PVVT+T
fprintf(dataFile,"%d,",Flag_PVVT_minus_T);      //PVVT-T
fprintf(dataFile,"%d,",Flag_P_minus_XCLT);      //P-XCLT
fprintf(dataFile,"%d,",Flag_P_minus_XFBT);      //P-XFBT
fprintf(dataFile,"%d,",Flag_P_minus_XPTP);              //P-XPTP
fprintf(dataFile,"%d,",Flag_P_minus_XTIT);              //P-XTIT
fprintf(dataFile,"%d,",Flag_P_minus_XTOT);      //P-XTOT
fprintf(dataFile,"%d,",Flag_P_minus_XVBT);      //P-XVBT

//
// Electrical Power Subsystem - FLAGS
//
fprintf(dataFile,"%d,",Flag_ECUSBI);            //ECUSBI
fprintf(dataFile,"%d,",Flag_ECUSBV);            //ECUSBV
fprintf(dataFile,"%d,",Flag_EED1AV);            //EED1AV
fprintf(dataFile,"%d,",Flag_EED1BV);            //EED1BV
fprintf(dataFile,"%d,",Flag_EED5AV);            //EED5AV
fprintf(dataFile,"%d,",Flag_EED5BV);            //EED5BV
fprintf(dataFile,"%d,",Flag_EPBA1T);            //EPBA1T
fprintf(dataFile,"%d,",Flag_EPBA2T);            //EPBA2T
```

```
fprintf(dataFile,"%d,",Flag_EPBB1T);          //EPBB1T
fprintf(dataFile,"%d,",Flag_EPBB2T);          //EPBB2T
fprintf(dataFile,"%d,",Flag_EPBC1T);          //EPBC1T
fprintf(dataFile,"%d,",Flag_EPBC2T);          //EPBC2T
fprintf(dataFile,"%d,",Flag_EPBSAI);                  //EPBSAI
fprintf(dataFile,"%d,",Flag_EPLP2T);          //EPLP2T
fprintf(dataFile,"%d,",Flag_EPSBAI);                  //EPSBAI
fprintf(dataFile,"%d,",Flag_EPSBAV);          //EPSBAV
fprintf(dataFile,"%d,",Flag_EPSBBI);                  //EPSBBI
fprintf(dataFile,"%d,",Flag_EPSBBV);          //EPSBBV
fprintf(dataFile,"%d,",Flag_EPSBCI);                  //EPSBCI
fprintf(dataFile,"%d,",Flag_EPSBCV);          //EPSBCV
fprintf(dataFile,"%d,",Flag_EPSDBV);          //EPSDBV
fprintf(dataFile,"%d,",Flag_EPSEBT);          //EPSEBT
fprintf(dataFile,"%d,",Flag_EPSEDT);          //EPSEDT
fprintf(dataFile,"%d,",Flag_EPSEET);          //EPSEET
fprintf(dataFile,"%d,",Flag_EPSLBI);                  //EPSLBI
fprintf(dataFile,"%d,",Flag_EPSPBV);          //EPSPBV
fprintf(dataFile,"%d,",Flag_EPUC1T);          //EPUC1T
fprintf(dataFile,"%d,",Flag_EPUC2T);          //EPUC2T
fprintf(dataFile,"%d,",Flag_EPUN1T);          //EPUN1T
fprintf(dataFile,"%d,",Flag_ESCLBI);                  //ESCLBI
fprintf(dataFile,"%d,",Flag_ESP31T);                  //EPS31T
fprintf(dataFile,"%d,",Flag_ESP32T);                  //EPS32T
fprintf(dataFile,"%d,",Flag_SLOART);          //SLOART
fprintf(dataFile,"%d,",Flag_SPXCPT);          //SPXCPT

//
//-----------------------------------------------------
//
// ** DISCRETES **
//
//-----------------------------------------------------
//

// Link 2 Communications Subsystem - Discretes

fprintf(dataFile,"%d,",State_C2ASPB);         //C2ASPB
fprintf(dataFile,"%d,",State_EDABRB);         //EDABRB
fprintf(dataFile,"%d,",State_EDBBRB);         //EDBBRB
fprintf(dataFile,"%d,",State_EDTUBB);         //EDTUBB
fprintf(dataFile,"%d,",State_EDTUAB);         //EDTUAB
fprintf(dataFile,"%d,",State_EDTUDB);         //EDTUDB
fprintf(dataFile,"%d,",State_EKG2AB);         //EKG2AB
fprintf(dataFile,"%d,",State_EKG2BB);         //EKG2BB
fprintf(dataFile,"%d,",State_ERBUDB);         //ERBUDB
```

```
fprintf(dataFile,"%d,",State_ET2AAB);             //ET2AAB
fprintf(dataFile,"%d,",State_ET2AEB);             //ET2AEB
fprintf(dataFile,"%d,",State_ET2AOB);             //ET2AOB
fprintf(dataFile,"%d,",State_ET2BAB);             //ET2BAB
fprintf(dataFile,"%d,",State_ET2BEB);             //ET2BEB
fprintf(dataFile,"%d,",State_ET2BOB);             //ET2BOB

//
// Link 1 Communications Subsystem - Discretes
//
fprintf(dataFile,"%d,",State_CDCAMB);             //CDCAMB
fprintf(dataFile,"%d,",State_CDCBMB);             //CDCBMB
fprintf(dataFile,"%d,",State_CSW1PB);             //CSW1PB
fprintf(dataFile,"%d,",State_CSW2PB);             //CSW2PB
fprintf(dataFile,"%d,",State_CSW3PB);             //CSW3PB
fprintf(dataFile,"%d,",State_CSW4PB);             //CSW4PB
fprintf(dataFile,"%d,",State_CSW5PB);             //CSW5PB
fprintf(dataFile,"%d,",State_E1ASOB);             //E1ASOB
fprintf(dataFile,"%d,",State_E1BSOB);             //E1BSOB
fprintf(dataFile,"%d,",State_EDC1AB);             //EDC1AB
fprintf(dataFile,"%d,",State_EDECAB);             //EDECAB
fprintf(dataFile,"%d,",State_EDECBB);             //EDECBB
fprintf(dataFile,"%d,",State_EKG1AB);             //EKG1AB
fprintf(dataFile,"%d,",State_EKG1BB);             //EKG1BB
fprintf(dataFile,"%d,",State_EP1AHB);             //EP1AHB
fprintf(dataFile,"%d,",State_EP1ASB);             //EP1ASB
fprintf(dataFile,"%d,",State_EP1BSB);             //EP1BSB
fprintf(dataFile,"%d,",State_EP1BHB);             //EP1BHB
fprintf(dataFile,"%d,",State_ET1AAB);             //ET1AAB
fprintf(dataFile,"%d,",State_ET1AEB);             //ET1AEB
fprintf(dataFile,"%d,",State_ET1AOB);             //ET1AOB
fprintf(dataFile,"%d,",State_ET1BOB);             //ET1BOB
fprintf(dataFile,"%d,",State_ET1BAB);             //ET1BAB
fprintf(dataFile,"%d,",State_ET1BEB);             //ET1BEB

//
// Propulsion Subsystem - Discretes
//

fprintf(dataFile,"%d,",State_ACJEAB);             //ACJEAB
fprintf(dataFile,"%d,",State_ACJEBB);             //ACJEBB
fprintf(dataFile,"%d,",State_ACTEAB);             //ACTEAB
fprintf(dataFile,"%d,",State_ACTEBB);             //ACTEBB
fprintf(dataFile,"%d,",State_AGGAEB);             //AGGAEB
fprintf(dataFile,"%d,",State_AGGBEB);             //AGGBEB
fprintf(dataFile,"%d,",State_ASJEAB);             //ASJEAB
```

```
fprintf(dataFile,"%d,",State_ASJEBB);          //ASJEBB
fprintf(dataFile,"%d,",State_AVVEAB);          //AVVEAB
fprintf(dataFile,"%d,",State_AVVEBB);          //AVVEBB

//
// Electrical Power Subsystem - Discretes
//

fprintf(dataFile,"%d,",State_EAACMB);          //EAACMB
fprintf(dataFile,"%d,",State_EADBYB);          //EADBYB
fprintf(dataFile,"%d,",State_EASBPB);          //EASBPB
fprintf(dataFile,"%d,",State_EASBRB);          //EASBRB
fprintf(dataFile,"%d,",State_EBACMB);          //EBACMB
fprintf(dataFile,"%d,",State_EBAHEB);          //EBAHEB
fprintf(dataFile,"%d,",State_EBAK1B);          //EBAK1B
fprintf(dataFile,"%d,",State_EBAK2B);          //EBAK2B
fprintf(dataFile,"%d,",State_EBAK3B);          //EBAK3B
fprintf(dataFile,"%d,",State_EBARDB);          //EBARDB
fprintf(dataFile,"%d,",State_EBARTB);          //EBARTB
fprintf(dataFile,"%d,",State_EBBHEB);          //EBBHEB
fprintf(dataFile,"%d,",State_EBBK1B);          //EBBK1B
fprintf(dataFile,"%d,",State_EBBK2B);          //EBBK2B
fprintf(dataFile,"%d,",State_EBBK3B);          //EBBK3B
fprintf(dataFile,"%d,",State_EBBRDB);          //EBBRDB
fprintf(dataFile,"%d,",State_EBBRTB);          //EBBRTB
fprintf(dataFile,"%d,",State_EBCHEB);          //EBCHEB
fprintf(dataFile,"%d,",State_EBCK1B);          //EBCK1B
fprintf(dataFile,"%d,",State_EBCK2B);          //EBCK2B
fprintf(dataFile,"%d,",State_EBCK3B);          //EBCK3B
fprintf(dataFile,"%d,",State_EBCRDB);          //EBCRDB
fprintf(dataFile,"%d,",State_EBCRTB);          //EBCRTB
fprintf(dataFile,"%d,",State_EBDBYB);          //EBDBYB
fprintf(dataFile,"%d,",State_EBVLSB);          //EBVLSB
fprintf(dataFile,"%d,",State_ECACMB);          //ECACMB
fprintf(dataFile,"%d,",State_ECDBYB);          //ECDBYB
fprintf(dataFile,"%d,",State_EDUVSB);          //EDUVSB
fprintf(dataFile,"%d,",State_ESP1PB);          //ESP1PB
fprintf(dataFile,"%d,",State_ESP1RB);          //ESP1RB
fprintf(dataFile,"%d,",State_ESP2PB);          //ESP2PB
fprintf(dataFile,"%d,",State_ESP2RB);          //ESP2RB
fprintf(dataFile,"%d,",State_ESP3PB);          //ESP3PB
fprintf(dataFile,"%d,",State_ESP3RB);          //ESP3RB
fprintf(dataFile,"%d,",State_ESP4PB);          //ESP4PB
fprintf(dataFile,"%d,",State_ESP4RB);          //ESP4RB
```

```
// ***********************************
// ** End of simulated satellite data **
// ***********************************



// *************************************************
// ** Begin Satellite Controller Performance Measures **
// *************************************************
//

fprintf(dataFile,"%d,",dcCursorX);          //cursor coordinate
fprintf(dataFile,"%d\n",dcCursorY);         //cursor coordinate

//
//      Note the existence of the '\n' in the format statement follwing
//      following the last 'fprint in this list'. For appropriate formatting, it
//      is necesary for the last fprint to have a '\n' as
//      shown. If additional variables are added to the list,
//      maintain the standard of having the last collected
//      variable, insert a '\n' as part of the format.

// Button pressed (included mouse click and actions caused by voice command)- TBD
// Voice Recognition Beginning Point (a flag?) - TBD
// Phrase Recognized (Can we capture the phrase that the speech recognizer
// decided the controller said?) - TBD
```

VIII. User Modifications

Screen Layouts

Previous chapters have dealt with the Altia displays, which consist of pass plans, electrical flow diagrams, and communication system flows. This chapter deals with the rest of the desktop real estate which is used by the Space Operation simulation.

All the displays which are not Altia derived, were created using the MS compiler. To modify the displays, the compiler is required as well. All three executables used with the Space Operations simulation (TimeClient.exe, SpaceOps.exe, and TimeServer.exe) are using desk top real estate, and have displays that can be changed. If the user should so desire it, the following procedure will help.

To change the experimenter station desktop display, activate the MS VC++ compiler by double clicking on the SpaceOps.dsw icon. This will activate the MS compiler, and load the source code. In the middle left of the view, there three or four 'view' tabs that can be used to select the class view, resource view, or file view. If not already selected, select the resource view by clicking with the mouse. Expand the SpaceOps resource folder, and expand the dialog folder that shows up. The desktop displays are shown as files in the expansion. Double clicking on a file will cause the resource to show up in the editing window. To change the look and feel of the desktop displays, use the mouse to select and manipulate the display.

Adding Additional Voice Commands

To add additional commands to the system, you need to modify the header file 'commands.h'. The syntax is complicated, so it is recommended that if you want to add a string to be recognized, it is best to copy an existing phrase, and change the text inside the quotes. The first argument to the m_VMenu.Add Dragon API, is a constant that is unique to the phrase being recognized. It needs to be created and added to 'voicedefine.h. Change the text (inside the double quotes in the second argument) to the required phrase. Having done made these changes, it is also necessary to add an additional case statement in messagemap.cpp. This is best done by copying anexisting case, and making the relevant changes. After the changes are made, it will be necessary to compile and link the TimeClient executable.

Appendix 6 – Experimenter User's Manual

TABLE OF CONTENTS

LIST OF FIGURES

Experimenter Workstation Description

This document describes the experimenter workstation in the satellite control test bed developed by Monterey Technologies, Inc. (MTI).

Figure 1 shows the display on the experimenter's workstation.

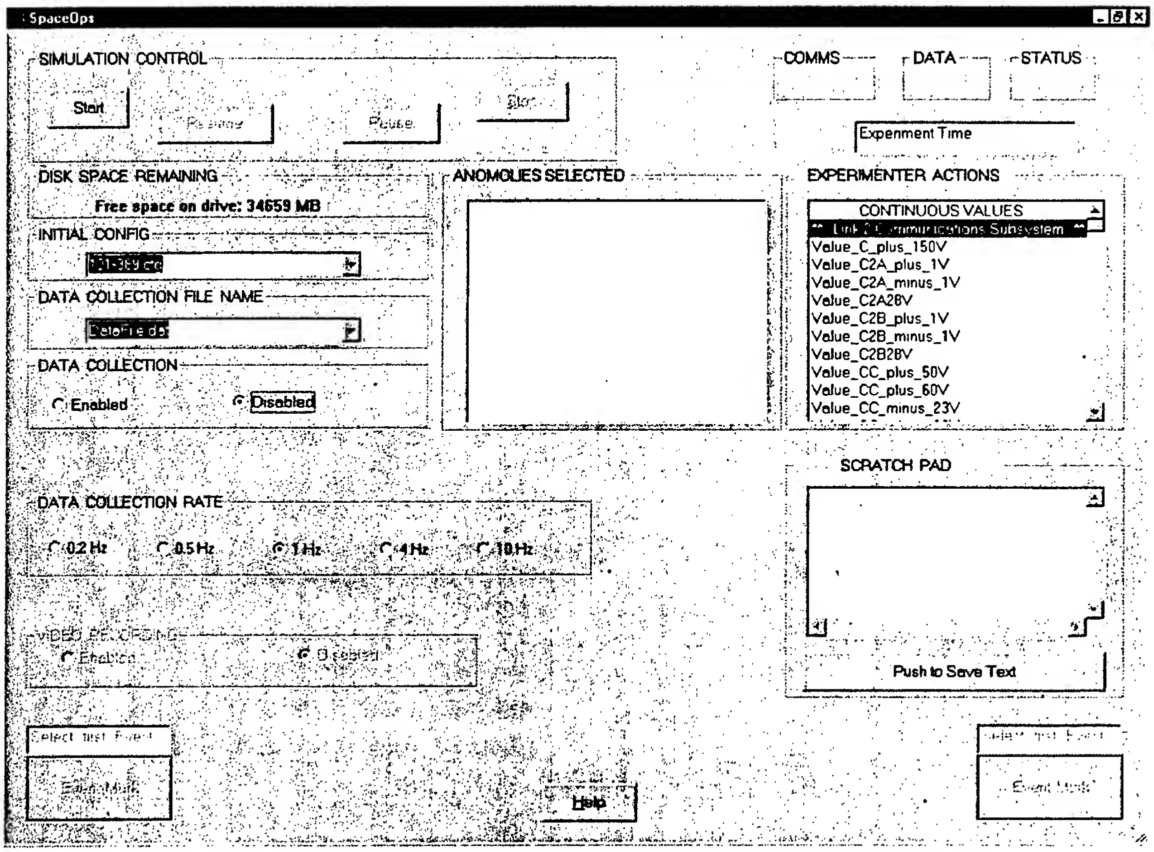


Figure 1. Experimenter Workstation With No Continuous or Discrete Anomalies Active.

DESCRIPTION OF EXPERIMENTER CONTROLS AND DISPLAYS

The experimenter controls are organized into functional groups. These areas are described here. The descriptions begin with the upper left part of the screen and proceed in a clockwise direction.

Simulation Control.

The simulation control functions are in the upper left portion of the screen. The functions available are START, RESUME, PAUSE, and STOP. The functions that are not available in the current state of the simulation are "grayed out". For example, in Figure 1 the simulation is not running. Therefore, RESUME, PAUSE, and STOP are not available. When the simulation is running, these functions become available.

<u>Comms.</u>
The COMMS box has an indication of the Ethernet communications in the simulation. This box contains a green field when the Ethernet communications are operating, and a red field when they are not operating. This indicator is only present when a simulation is running.

<u>Data.</u>
The DATA field contains a green field when data is being collected and a red field when data is not being collected. This indicator is only present when a simulation is running.

<u>Status.</u>
TBD

<u>Experiment Time.</u>
This field displays the time being used by the simulation. This time is in Coordinated Universal Time (UTC, aka Zulu time or Greenwich Mean Time [GMT]). This is set by the experimenter for each simulation session. This allows the use of tasks that must be performed at specific times. Firing of thrusters to make adjustments to the orbit or orientation of the satellite is an example of such a task.

<u>Experimenter Actions.</u>
The EXPERIMENTER ACTIONS area is used by the controller to select and deselect anomalies. This process is discussed in greater detail below.

<u>Scratch Pad.</u>
The SCRATCH PAD area allows the experimenter to type in comments. In order to type in text the experimenter must first make the field active by putting the cursor in the field. This can be done by pressing on the field (touch screen) or by placing the cursor in the field and clicking (mouse). These comments are added to the data file being created for that simulation session. The time code on the text is based on the time where the experimenter presses, or clicks on, the "Push To Save Text" button located below the SCRATCH PAD.

<u>Event Mark.</u>
The EVENT MARK button appears in the lower left and lower right corners of the display. The experimenter can press or click on either button to enter an event marker into the data file. The event markers are integers. The event markers are placed in the data file along with the other data collected during that frame.

Immediately above each EVENT MARK button is a window that displays the number of the last marker entered. For example, before any event markers have been entered this box will show 0 (zero). The first time the experimenter presses the EVENT MARK button the number increments so a 1 (one) is displayed.

Event markers are used to make it easy for the experimenter to find portions of the data file of interest.

Help.
No on-line help has been implemented in the test bed.

Video Recording.
This filed is not used in the current system. It was placed in the experimenter display to allow for future expansion of the system.

Data Collection Rate.
This filed is used to select the rate of data collection. Five rates are currently available; 0.2, 0.5, 1.0, 4, and 10 Hz. These correspond to 5 sec, 2 sec, 1 sec, 250 msec, and 100 msec between data samples. The experimenter selects the data collection rate that best suits the need for temporal resolution and minimizes the size of the resulting data file.

Note that this is selected prior to the start of a simulation session, and may not be reset during a session.

Data Collection.
This control is used to turn data collection on and off. In the off position no data file is created. In the on position a data file is created for that session.

Data Collection File Name.
The experimenter may enter a unique file name in this field. If the default name is not changed, then the experimenter would need to rename the file after the session in order to avoid having the file over-written during a subsequent session.

Initial Configuration.
This filed is used by the experimenter to select among any pre-created scenarios.

Disk Space Remaining.
This field show the amount of free disk space available for data storage. The experimenter can use this information to determine if sufficient space is available to store the data that will be generated during a session. If insufficient space is available, then the experimenter needs to free up disk space before collecting data.

Anomalies Selected.
The ANOMALIES SELECTED filed displays the anomalies that are currently active, and the level of those anomalies. This is a display only field; the experimenter cannot manipulate this information directly.

SELECTING, ACTIVATING, AND DEACTIVATING ANOMALIES

The experimenter uses the EXPERIMENTER ACTIONS field to select and deselect anomalies. All of the continuous and discontinuous variables being simulated have anomalies available.

Continuous Variables.

For continuous variables, the experimenter may select one of four anomaly levels. These levels are:

Caution – Low,
Warning – Low,
Warning – High, and
Caution – High.

Here, "Low" indicates that the value is less than the lower boundary of the nominal range and "High" is greater than the upper boundary of the nominal range. Warnings are in the yellow range and Cautions are in the red range.

<u>Selecting And Activating An Anomaly On A Continuous Variable</u>
In order to select an anomaly the experimenter must select the variable and the level of the anomaly. As an example, suppose that the experimenter would like the value of CCT1AW to be in the Warning – Low range. This process is described here in a step by step manner.

First, the experimenter needs to locate the variable. CCT1AW is in the Link 1 subsystem. (Four subsystems of the DSP satellite are simulated. These subsystems are Link 1, Link 2, Propulsion, and Electrical.) The variable is selected by scrolling down the EXPERIMENTER ACTIONS field until the variable is in the display. Figure 2 shows the display after the experimenter has scrolled down until CCT1AW is in the display

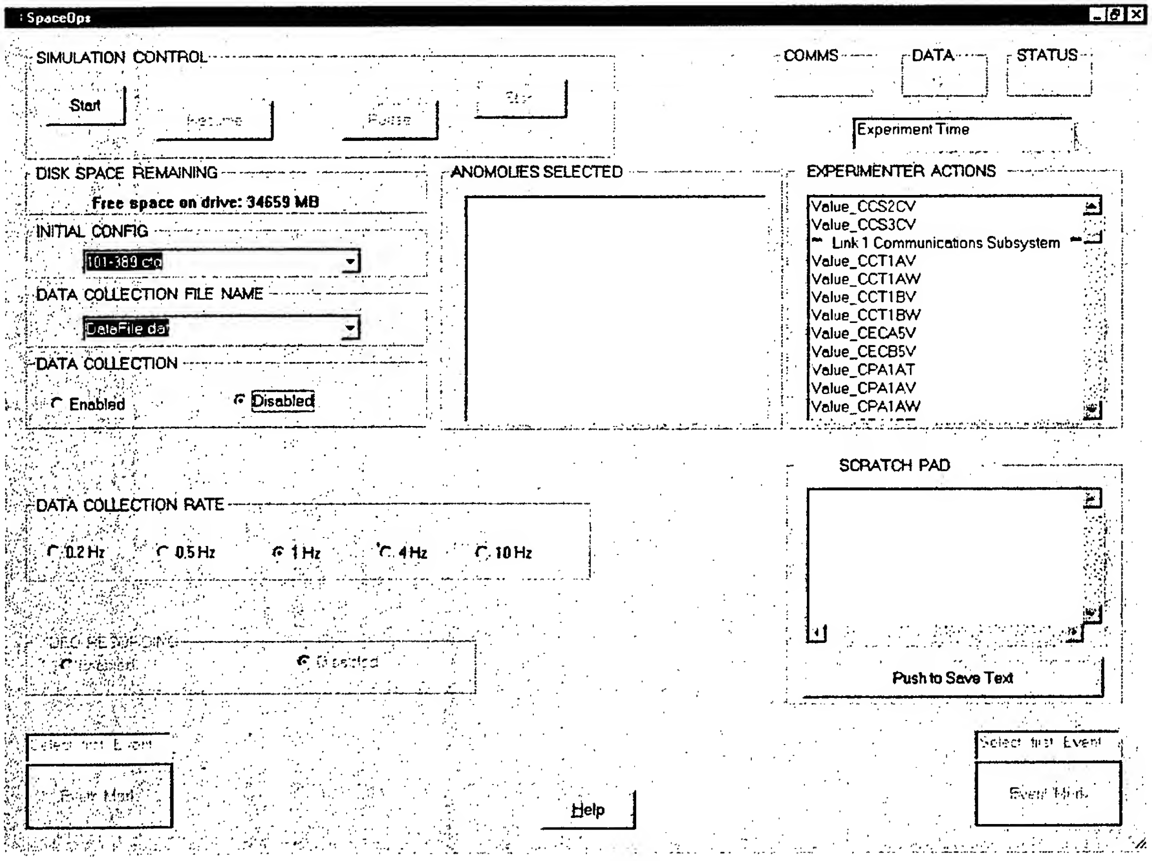


Figure 2. Experimenter Workstation Showing The Desired Continuous Variable In The Experimenter Actions Field.

Once the desired variable is in the display, the experimenter clicks on the name. Clicking on the variable name will cause a pop up window to be displayed. Figure 3 shows the pop up window

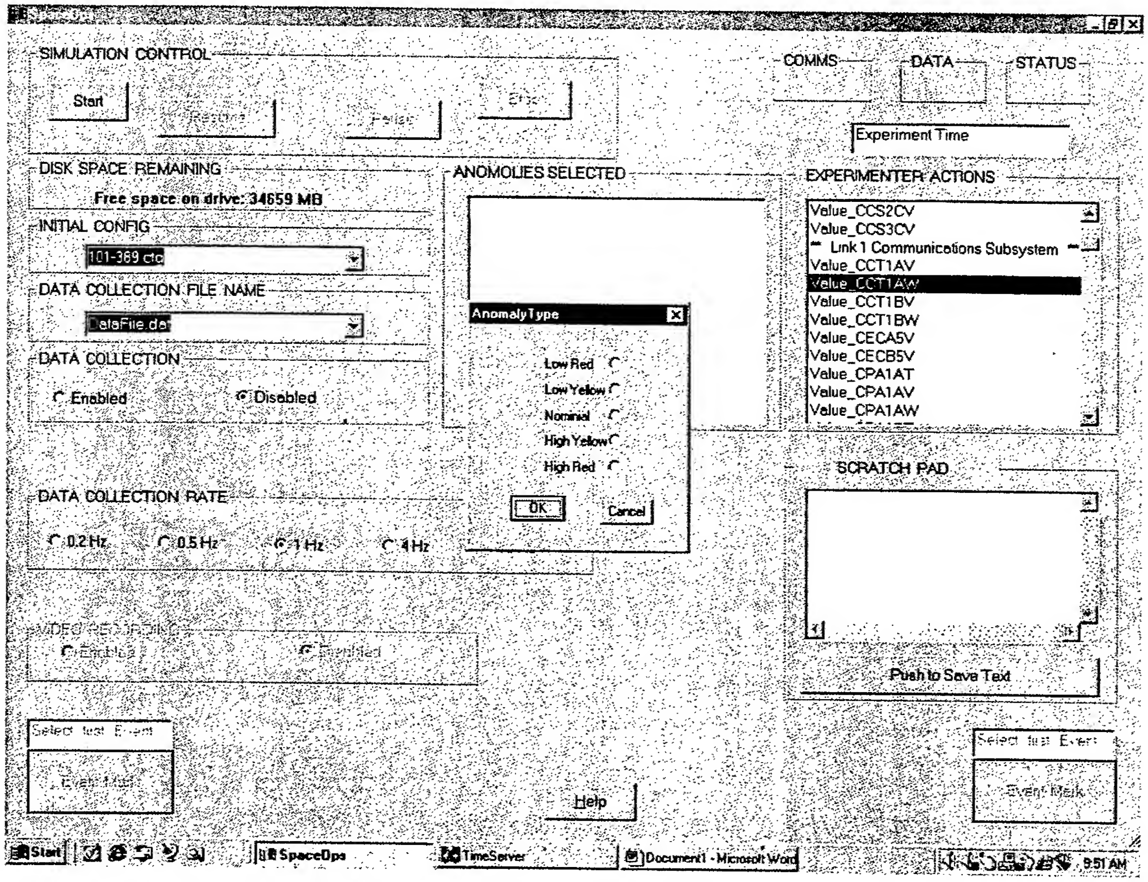


Figure 3. The Experimenter's Display After Selecting CCT1AW For An Anomaly.

The experimenter than selects the type of anomaly by clicking on the desired level. In this example, the experimenter wishes to have a Warning-Low anomaly so she clicks on the Low Yellow button. This is shown in Figure 4.

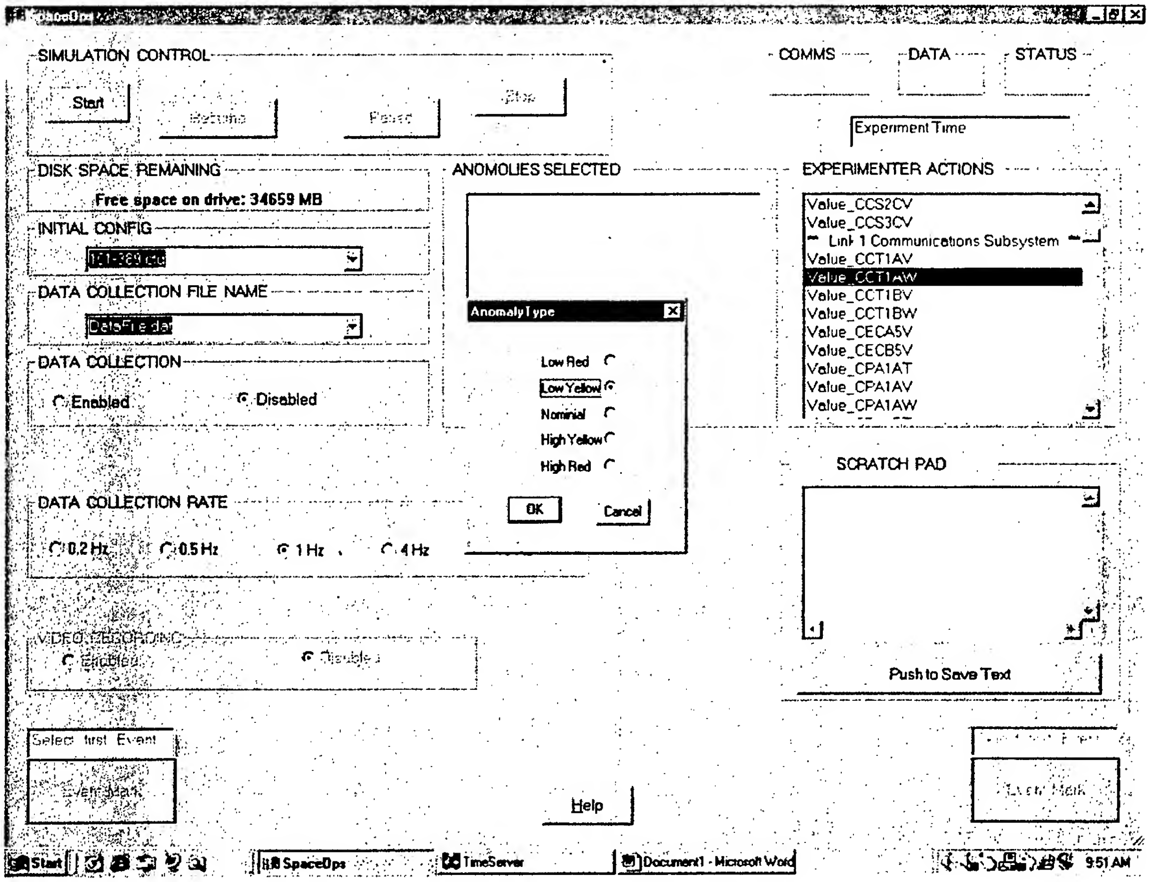


Figure 4. The Experimenter's Display Showing A "Low-Yellow" Anomaly.

The experimenter activates the anomaly by clicking on the OK button in the pop up window. The experimenter could cancel the action by pressing the CANCEL button. This would cause the pop up window to be removed from the display and the anomaly would NOT be made active or changed.

When the experimenter clicks OK, the pop up window is removed from the display and the name of the variable containing the anomaly appears in the ANOMALIES SELECTED window. Figure 5 shows the display with the anomaly active.

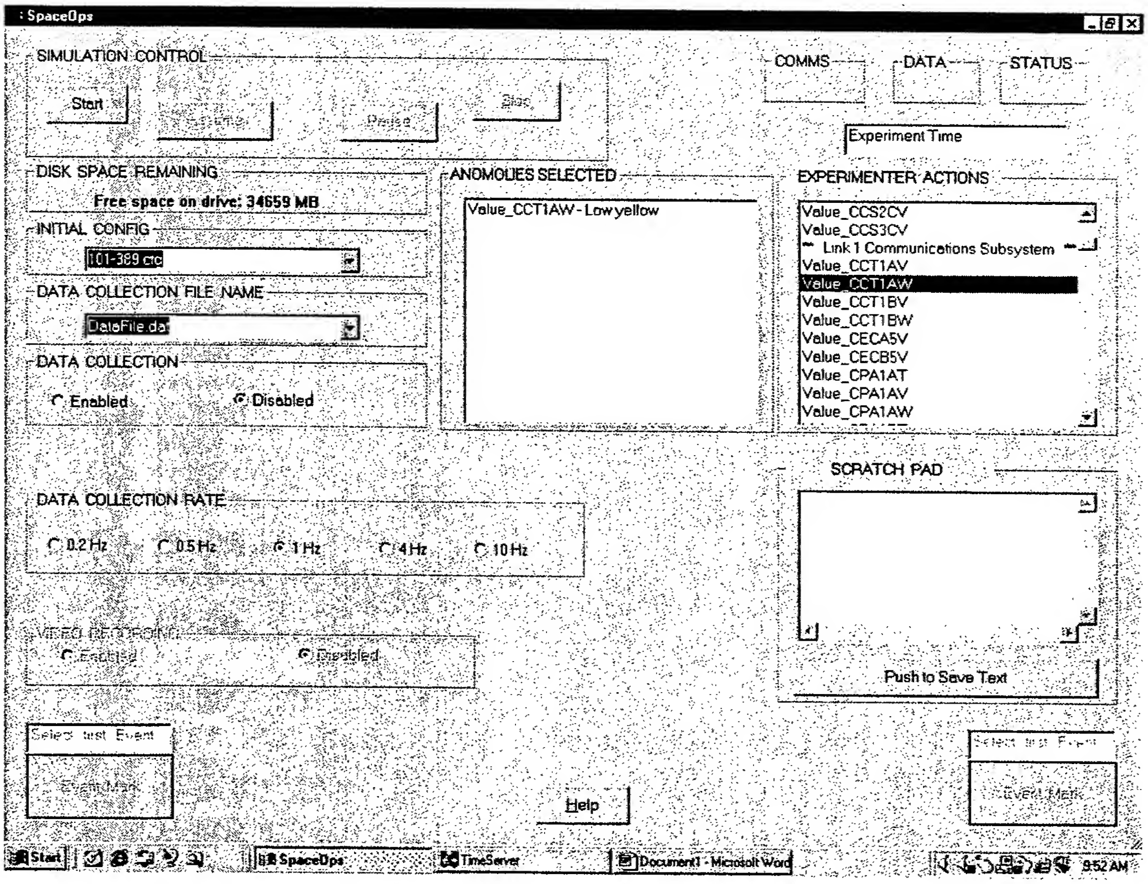


Figure 5. The Experimenter's Display After Activating An Anomaly On A Continuous Variable.

When one or more anomalies are selected, the state of health indicator on the controllers display will reflect the fact.

<u>Deactivating An Anomaly On A Continuous Variable.</u>
The process of removing an anomaly is identical to the process for activating an anomaly.

First, the controller scrolls down in the EXPERIMENTER ACTIONS field until the variable is displayed. The display shown in Figure 5 shows an active anomaly for CCT1AW, and the variable name being displayed.

The controller then clicks on the variable name in the EXPERIMENTER ACTIONS field. A pop up window will then be displayed. The experimenter then clicks on "Nominal". This is shown in Figure 6.

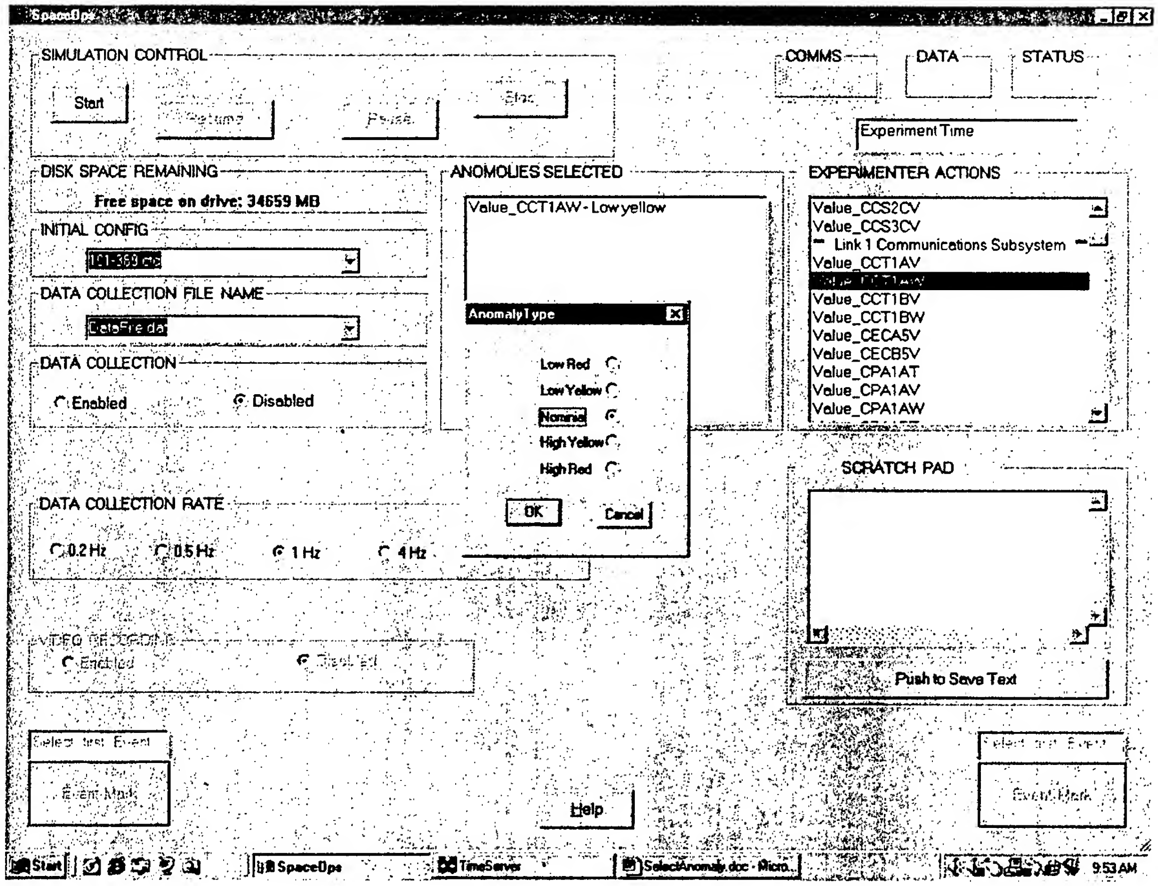


Figure 6. Deactivating An Anomaly On A Continuous Variable.

The controller then clicks on the OK button to remove the anomaly. (The controller could also cancel the action by clicking the Cancel button at this time.) Once the controller clicks on the OK button, the pop up window is removed from the display and the variable is removed from the list in the ANOMALIES SELECTED window. Figure 7 shows the display with the anomaly removed.

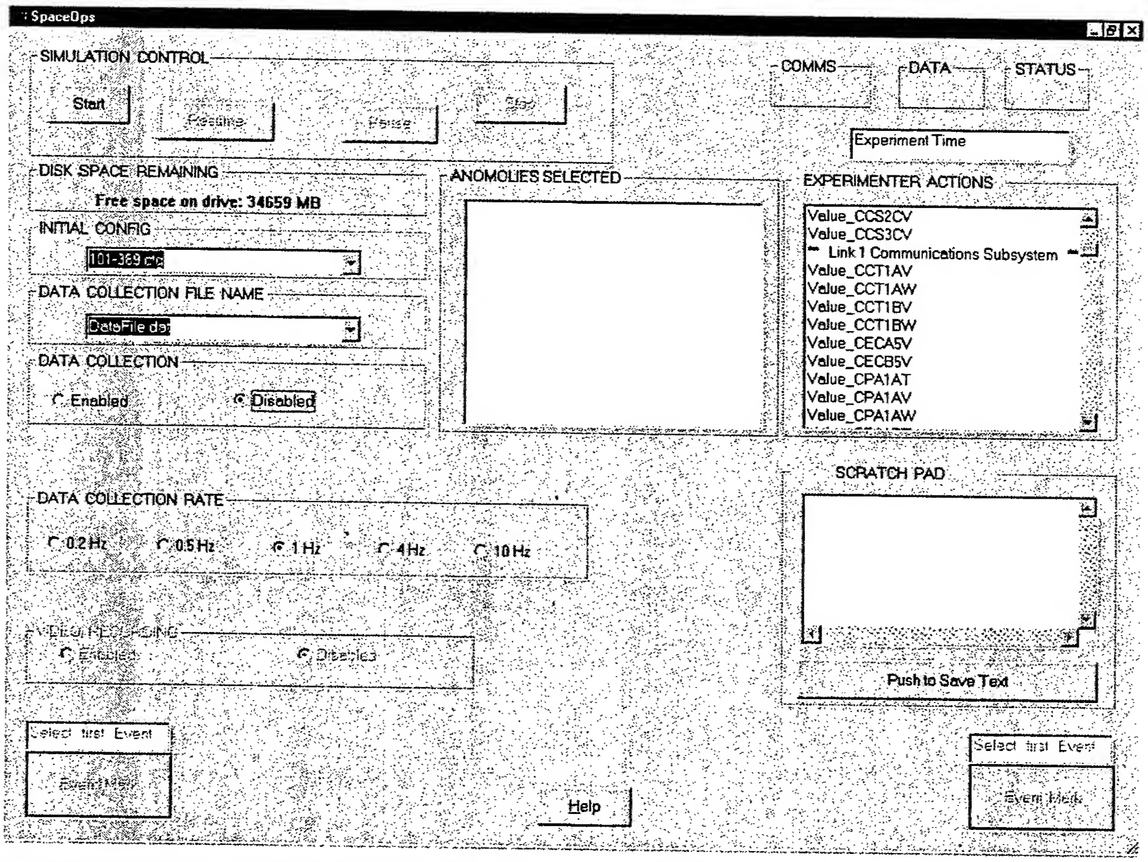


Figure 7. The Experimenter's Display After Deactivating An Anomaly.

If the anomaly removed was the only anomaly active, or if the anomaly removed is the only one in the warning or caution range, then the state of health display on the controller's workstation will be updated.

Discrete Variables.

The process of selecting and activating an anomaly for a discrete variable is almost identical to that used for selecting anomalies for continuous variables. The changes are (1) there are no levels of the anomaly to be selected from in the pop-up window, and (2) selection of an anomaly does not alter the controller's state of health display. It is worth noting that the effect of an anomaly on a discrete variable

<u>Selecting and Activating An Anomaly On A Discrete Variable</u>

Consider a case where the experimenter wishes to create an anomaly for the discrete variable C2ASPB. This variable is in the Link 2 Communications subsystem.

First, the experimenter scrolls through the variable list in the Experimenter Actions window until the desired variable is visible. Figure 8 shows the window when C2ASPB is displayed.

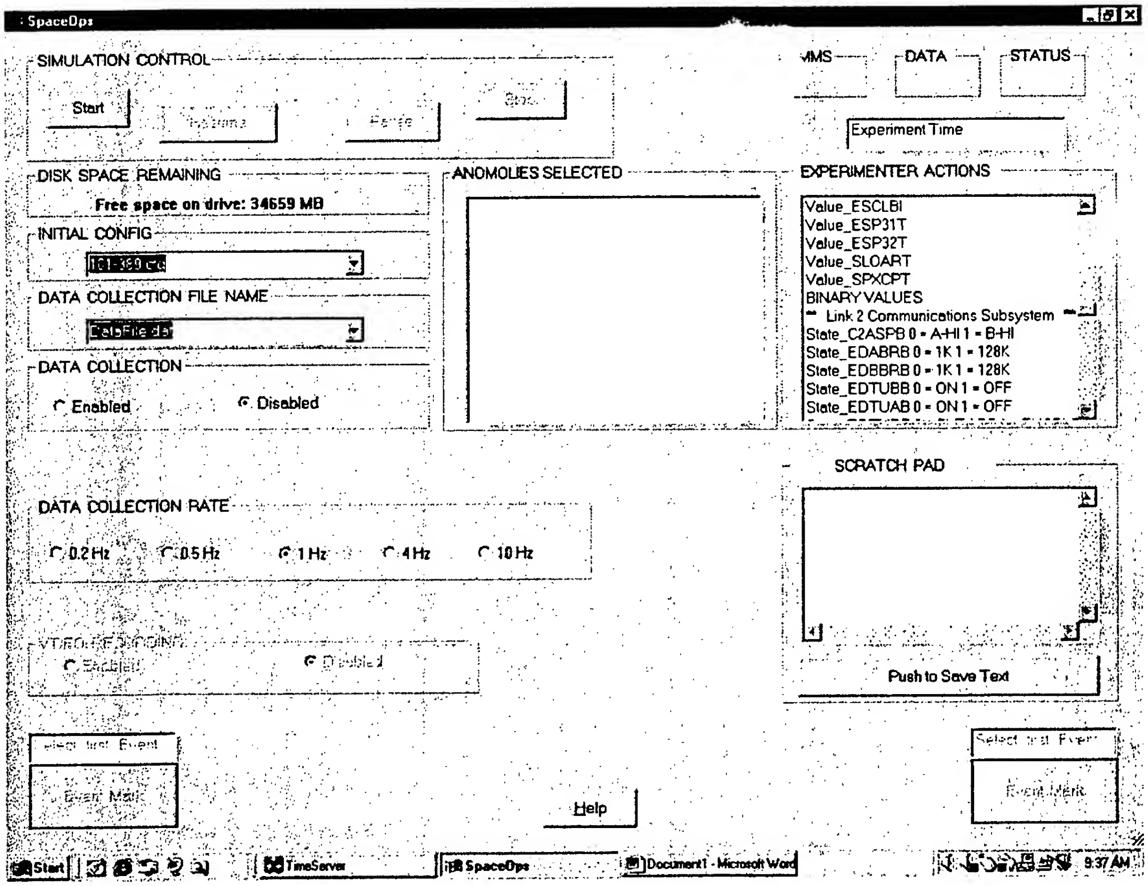


Figure 8. The Experimenter's Display Showing The Discrete Variable That Is To Have An Anomaly (C2ASPB).

When the desired variable is located, right click on the variable name. When the experimenter clicks on the variable name a pop-up window will be displayed. Figure 9 shows the display with the pop-up window.

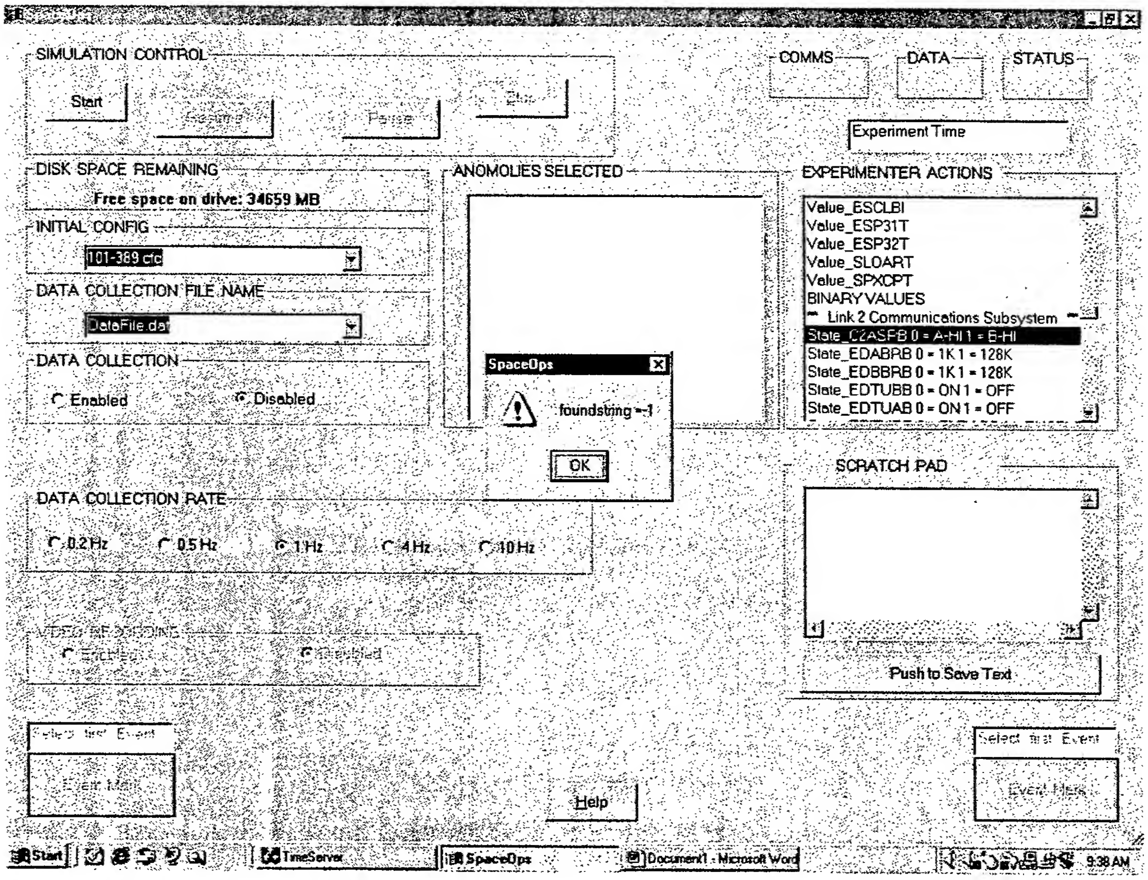


Figure 9. The Experimenter's Display After Selecting A Discrete Variable For An Anomaly, But Before Activating The Anomaly.

In order to activate the anomaly click on the "OK" button in the pop-up window. When "OK" is clicked the pop-up window is removed from the display and the anomaly is listed in the "Anomalies Selected" window. Figure 10 shows the display with the anomaly selected.

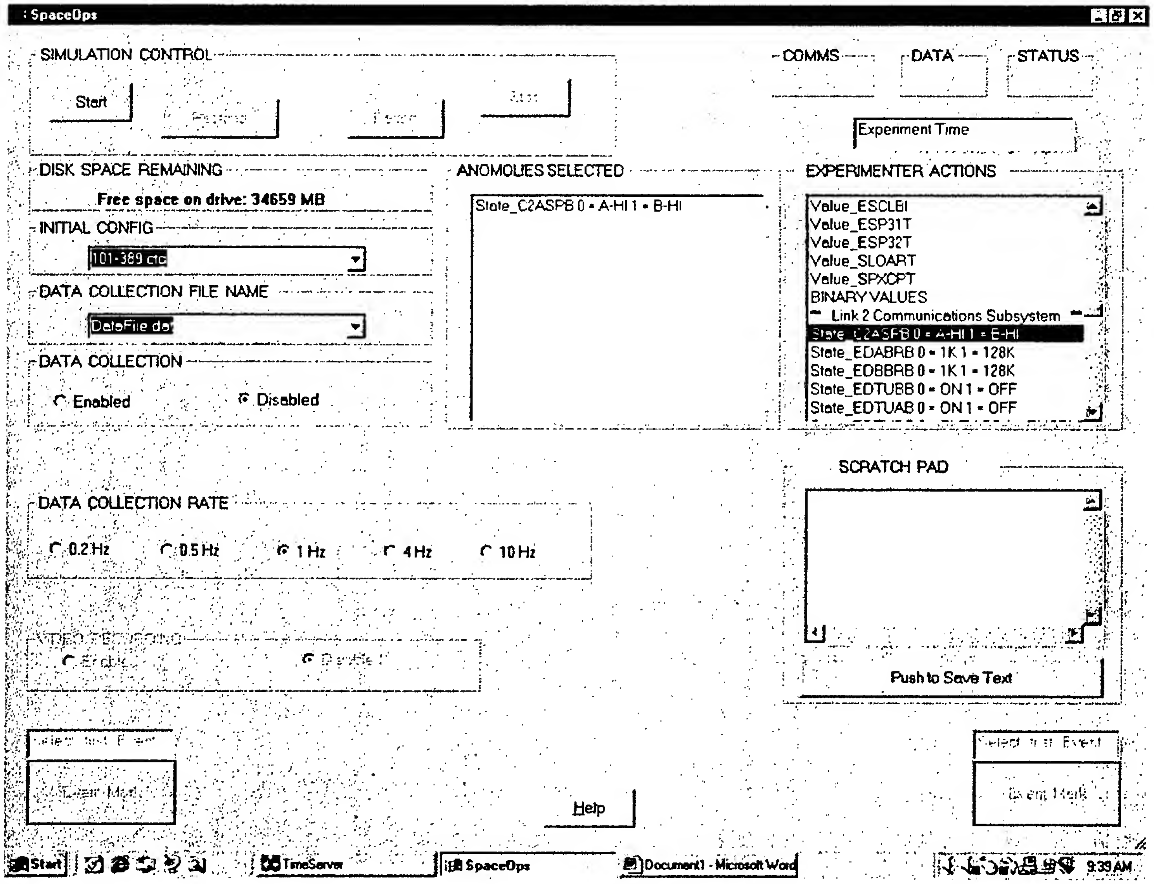


Figure 10. The Experimenter's Display With An Anomaly On A Discrete Variable Active.

<u>Deactivating An Anomaly On A Discrete Variable</u>
To deactivate an anomaly on a discrete variable it must be removed from the Anomalies Selected window. The first step in deactivating an anomaly is to scroll the Experimenter Actions window until the name of the variable is visible. Figure 11 shows the experimenter's display with C2ASPB visible.

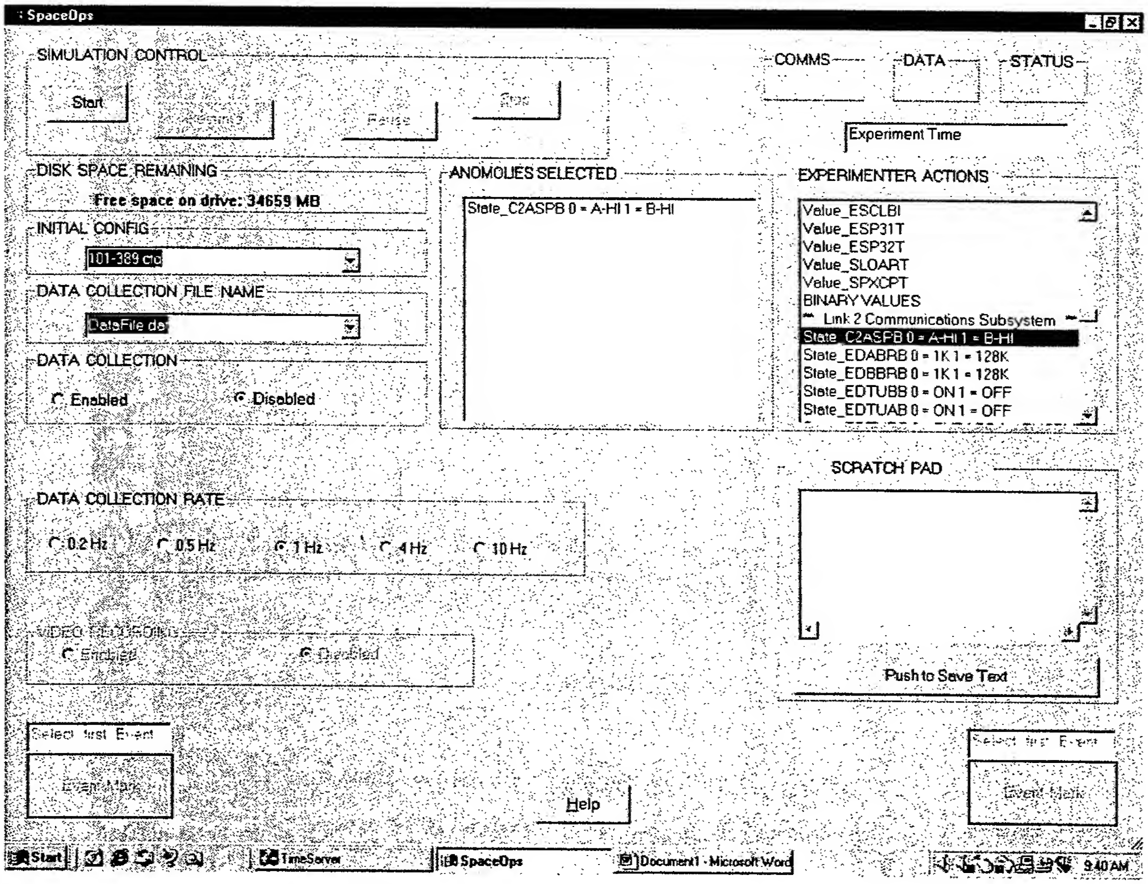


Figure 11. The Experimenter's Display Showing The Variable To Have An Anomaly Deactivated In The "Experimenter Actions" Window.

Once the desired variable name is visible in the Experimenter Actions window click on it. This will bring up a pop-up window. Figure 12 shows the display with the pop-up window visible.

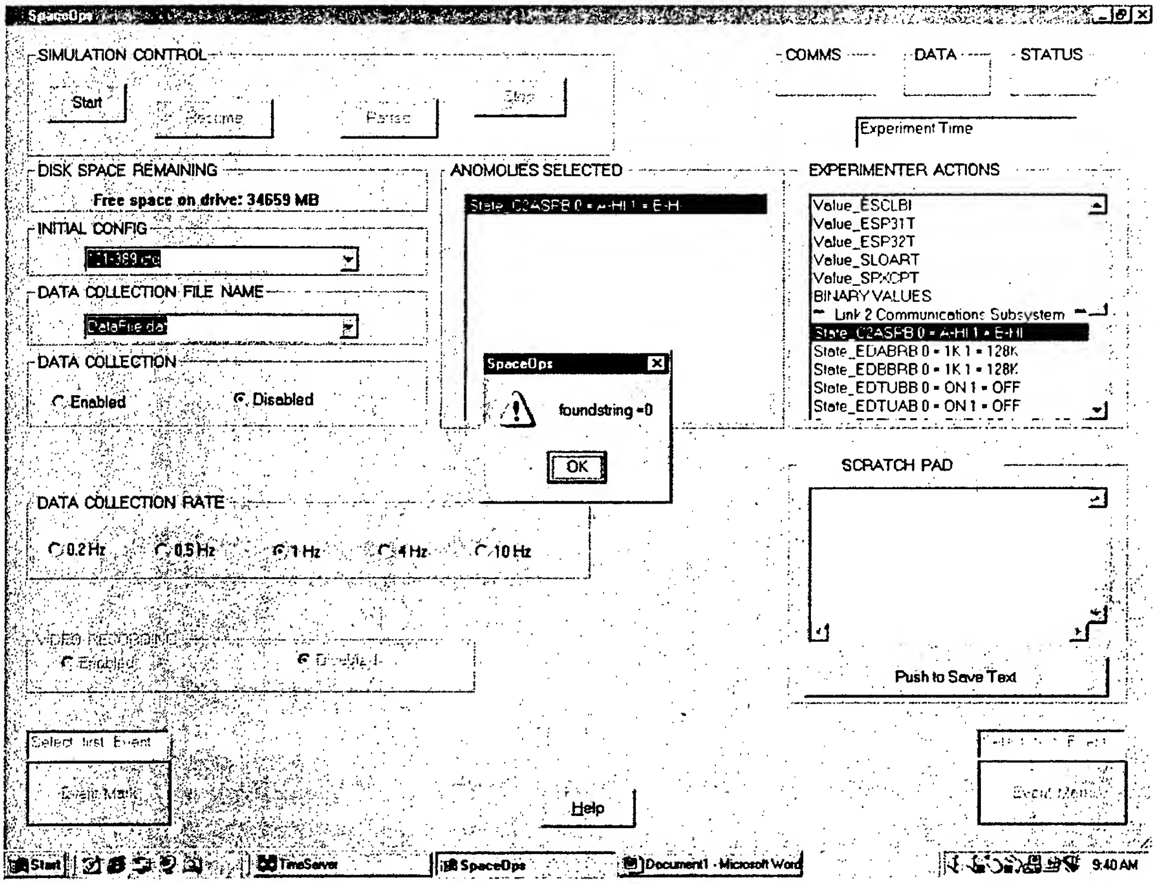


Figure 12. The Experimenter's Display Showing The Anomaly Ready To Be Deactivated.

Click on OK in the pop-up window. This will remove the anomaly from the active list. Figure 13 shows the experimenter's display with the discrete anomaly removed.

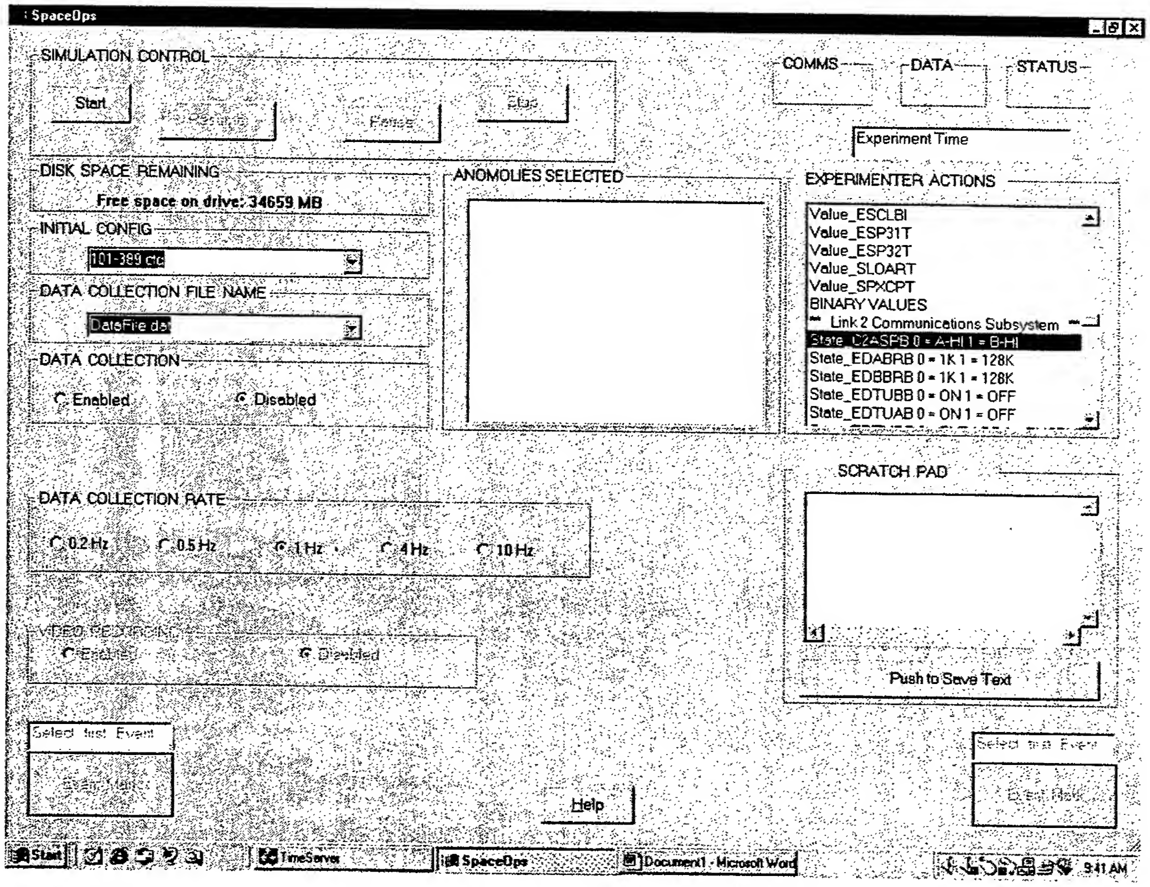


Figure 13. The Experimenter's Display After The Anomaly Has Been Deactivated.

DATA COLLECTION LIST

The Space Operations simulation will collect all of the variable defined in the file DataCollectionList.cpp, subject to the rules of the C++ language syntax. The path to this file is *d:\PostBTI\TimeClient\DataCollectionList.cpp.* As of this writing, a copy of DataCollectionList.cpp is shown in Table 1, below. If a specific variable is not required to be in the list of collected variables, it can be commented out in DataCollectionList.cpp. If it is commented out, after the TimeClient executable is recompiled, that specific variable will no longer be collected.

Data collection needs to be manually turned on with the experimenters interface (SpaceOps.exe). Default is for data collection to be off. If so desired, turn data collection on, select a data rate. The default data rate is 1Hz. Four other rates are selectable by the operator, .5Hz, .2Hz, 4Hz, and 10Hz. The only restriction on the amount of data collected, is the amount of disk space remaining on the ControlPoint computer. There are roughly 360 variables in the default collection list. If none are commented out, and the data rate is set to the fastest rate, the data collection code will collect at the rate of 51MB/hour.

This file can be edited using a standard text editor. However, the file must be a text file. This means that if a word processing package is used the file must be "saved as" a text file. Most, if not all, word processors have this capability. (If you save the file in Word format rather than as a text file, for example, data collection <u>will not</u> operate properly.)

This text file contains the names of all of the variables that can be collected during a simulated support. Since not every variable is of interest during every experiment, the experimenter has the ability to select the variables that are collected from the set of possibilities. Selecting a variable to be collected simply requires that the variable not be "commented out" in the list. For example, if the list was:

```
fprintf(dataFile,"%d,",dcMode);              // start/stop/pause/resume
fprintf(dataFile,"%d,",dcEventCount);              // operator entered
//fprintf(dataFile,"%d,",dcAnomalyFlag);       // operator entered
fprintf(dataFile,"%d,",dcAnomalyType);       // operator entered
```

```
Variable A
Variable B
// Variable C
Variable D
```

Then variables *dcMode*, *dcEventCount*, and *dcAnomalyType* would be collected. *dcAnomalyFlag* would not be collected as the double slashes (i.e., "//") preceding the variable name "comment out" this line.

If collecting *dcAnomalyFlag* is desired, then simple delete the double slashes. The data collection list would then appear as:

```
fprintf(dataFile,"%d,",dcMode);             // start/stop/pause/resume
fprintf(dataFile,"%d,",dcEventCount);       // operator entered
fprintf(dataFile,"%d,",dcAnomalyFlag);      // operator entered
fprintf(dataFile,"%d,",dcAnomalyType);      // operator entered
```

Table 1. Data Collection Variable List

```
// ***************************************************
// ** Data Collection for Satellite Control Test bed **
// ***************************************************
        fprintf(dataFile,"%d,",dcMode);             // start/stop/pause/resume
        fprintf(dataFile,"%d,",dcEventCount);       // operator entered
        fprintf(dataFile,"%d,",dcAnomalyFlag);      // operator entered
        fprintf(dataFile,"%d,",dcAnomalyType);      // operator entered
        fprintf(dataFile,"%s,",dcPseudoTime);       // experiment time
        fprintf(dataFile,"%s,",dcGmt);              // Greenwhich Mean Time
        fprintf(dataFile,"%d,",dcTime);             // wall time in integer format
        fprintf(dataFile,"%d,",dcExpTicks);         // 100 ms ticks past start
        fprintf(dataFile,"%d,",dcCount);            // simulation loop count

// *******************************************************************
// ** Data from the "thin simulation" of the satellite"            **
// **                                                               **
// ** The variable names, values and ranges were provided to MTI by CERES. **
// ** The satellite simulated is a DSCS bird.                      **
// **                                                               **
// ** The variables are grouped by subsystem.  Within each subsystem the **
// ** variables are listed in groups of five to improve legibility. **
// *******************************************************************

//--------------------------------------------------
//
// ** CONTINUOUS SATELLITE VARIABLES - RAW VALUES **
//
//--------------------------------------------------
```

```
//
//
// Link 2 Communications Subsystem - Continuous Variables - Raw Values
//
fprintf(dataFile,"%3.5f,",Value_C_plus_150V);          //C+150V
fprintf(dataFile,"%3.5f,",Value_C2A_plus_1V);          //C2A+1V
fprintf(dataFile,"%3.5f,",Value_C2A_minus_1V);         //C2A-1V
fprintf(dataFile,"%3.5f,",Value_C2A28V);               //C2A28V
fprintf(dataFile,"%3.5f,",Value_C2B_plus_1V);          //C2B+1V
fprintf(dataFile,"%3.5f,",Value_C2B_minus_1V);         //C2B-1V
fprintf(dataFile,"%3.5f,",Value_C2B28V);               //C2B28V
fprintf(dataFile,"%3.5f,",Value_CC_plus_50V);          //CC+150V
fprintf(dataFile,"%3.5f,",Value_CC_plus_60V);          //CC+60V
fprintf(dataFile,"%3.5f,",Value_CC_minus_23V);         //CC-23V
fprintf(dataFile,"%3.5f,",Value_CC_minus_60V);         //CC-60V
fprintf(dataFile,"%3.5f,",Value_CCINIT);               //CCINIT
fprintf(dataFile,"%3.5f,",Value_CCS1CV);               //CCS1CV
fprintf(dataFile,"%3.5f,",Value_CCS2CV);               //CCS2CV
fprintf(dataFile,"%3.5f,",Value_CCS3CV);               //CCS3CV
//
// Link 1 Communications Subsystem - Continuous Variables - Raw Values
//
fprintf(dataFile,"%3.5f,",Value_CCT1AV);               //CCT1AV
fprintf(dataFile,"%3.5f,",Value_CCT1AW);               //CCT1AW
fprintf(dataFile,"%3.5f,",Value_CCT1BV);               //CCT1BV
fprintf(dataFile,"%3.5f,",Value_CCT1BW);               //CCT1BW
fprintf(dataFile,"%3.5f,",Value_CECA5V);               //CECA5V
fprintf(dataFile,"%3.5f,",Value_CECB5V);               //CECB5V
fprintf(dataFile,"%3.5f,",Value_CPA1AT);               //CPA1AT
fprintf(dataFile,"%3.5f,",Value_CPA1AV);               //CPA1AV
fprintf(dataFile,"%3.5f,",Value_CPA1AW);               //CPA1AW
```

```
fprintf(dataFile,"%3.5f,",Value_CPA1BT);          //CPA1BT
fprintf(dataFile,"%3.5f,",Value_CPA1BV);          //CPA1BV
fprintf(dataFile,"%3.5f,",Value_CPA1BW);          //CPA1BW
fprintf(dataFile,"%3.5f,",Value_CT1AMT);          //CT1AMT
fprintf(dataFile,"%3.5f,",Value_CT1BMT);          //CT1BMT
//
// Propulsion Subsystem - Continuous Variables - Raw Values
//
fprintf(dataFile,"%3.5f,",Value_GGPRES);          //GGPRES
fprintf(dataFile,"%3.5f,",Value_LVLSEL);          //LVLSEL
fprintf(dataFile,"%3.5f,",Value_P_plus_XFBT);     //P+XFBT
fprintf(dataFile,"%3.5f,",Value_P_plus_XPTP);     //P+XPTP
fprintf(dataFile,"%3.5f,",Value_P_plus_XTOT);     //P+XTOT
fprintf(dataFile,"%3.5f,",Value_P_plus_XVBT);     //P+XVBT
fprintf(dataFile,"%3.5f,",Value_PFDIVT);          //PFDIVT.
fprintf(dataFile,"%3.5f,",Value_PGGFPT);          //PGGFPT
fprintf(dataFile,"%3.5f,",Value_PHLT_plus_T);     //PHLT+T
fprintf(dataFile,"%3.5f,",Value_PHLT_minus_T);    //PHLT-T
fprintf(dataFile,"%3.5f,",Value_PHT_plus_HT);     //PHT+HT
fprintf(dataFile,"%3.5f,",Value_PHT_minus_HT);    //PHT-HT
fprintf(dataFile,"%3.5f,",Value_PPLN1P);          //PPLN1P.
fprintf(dataFile,"%3.5f,",Value_PPLN2P);          //PPLN2P
fprintf(dataFile,"%3.5f,",Value_PVT_plus_HT);     //PVT+HT
fprintf(dataFile,"%3.5f,",Value_PVT_minus_HT);    //PVT-HT
fprintf(dataFile,"%3.5f,",Value_PVVT_plus_T);     //PVVT+T
fprintf(dataFile,"%3.5f,",Value_PVVT_minus_T);    //PVVT-T
fprintf(dataFile,"%3.5f,",Value_P_minus_XCLT);    //P-XCLT
fprintf(dataFile,"%3.5f,",Value_P_minus_XFBT);    //P-XFBT
fprintf(dataFile,"%3.5f,",Value_P_minus_XPTP);    //P-XPTP
fprintf(dataFile,"%3.5f,",Value_P_minus_XTIT);    //P-XTIT
fprintf(dataFile,"%3.5f,",Value_P_minus_XTOT);    //P-XTOT
```

```
fprintf(dataFile,"%3.5f,",Value_P_minus_XVBT);          //P-XVBT
//
// Electrical Power Subsystem - Continuous Variables - Raw Values
//
fprintf(dataFile,"%3.5f,",Value_ECUSBI);                 //ECUSBI
fprintf(dataFile,"%3.5f,",Value_ECUSBV);                 //ECUSBV
fprintf(dataFile,"%3.5f,",Value_EED1AV);                 //EED1AV
fprintf(dataFile,"%3.5f,",Value_EED1BV);                 //EED1BV
fprintf(dataFile,"%3.5f,",Value_EED5AV);                 //EED5AV
fprintf(dataFile,"%3.5f,",Value_EED5BV);                 //EED5BV
fprintf(dataFile,"%3.5f,",Value_EPBA1T);                 //EPBA1T
fprintf(dataFile,"%3.5f,",Value_EPBA2T);                 //EPBA2T
fprintf(dataFile,"%3.5f,",Value_EPBB1T);                 //EPPB1T
fprintf(dataFile,"%3.5f,",Value_EPBB2T);                 //EPBB2T
fprintf(dataFile,"%3.5f,",Value_EPBC1T);                 //EPBC1T
fprintf(dataFile,"%3.5f,",Value_EPBC2T);                 //EPBC2T
fprintf(dataFile,"%3.5f,",Value_EPBSAI);                 //EPBSAI
fprintf(dataFile,"%3.5f,",Value_EPLP2T);                 //EPLP2T
fprintf(dataFile,"%3.5f,",Value_EPSBAI);                 //EPSBAI
fprintf(dataFile,"%3.5f,",Value_EPSBAV);                 //EPSBAV
fprintf(dataFile,"%3.5f,",Value_EPSBBI);                 //EPSBBI
fprintf(dataFile,"%3.5f,",Value_EPSBBV);                 //EPSBBV
fprintf(dataFile,"%3.5f,",Value_EPSBCI);                 //EPSBCI
fprintf(dataFile,"%3.5f,",Value_EPSBCV);                 //EPSBCV
fprintf(dataFile,"%3.5f,",Value_EPSDBV);                 //EPSDBV
fprintf(dataFile,"%3.5f,",Value_EPSEBT);                 //EPSEBT
fprintf(dataFile,"%3.5f,",Value_EPSEDT);                 //EPSEDT
fprintf(dataFile,"%3.5f,",Value_EPSEET);                 //EPSEET
fprintf(dataFile,"%3.5f,",Value_EPSLBI);                 //EPSLBI
fprintf(dataFile,"%3.5f,",Value_EPSPBV);                 //EPSPBV
fprintf(dataFile,"%3.5f,",Value_EPUC1T);                 //EPUC1T
```

```
fprintf(dataFile,"%3.5f,",Value_EPUC2T);                //EPUC2T
fprintf(dataFile,"%3.5f,",Value_EPUN1T);                //EPUN1T
fprintf(dataFile,"%3.5f,",Value_ESCLBI);                //ESCLBI
fprintf(dataFile,"%3.5f,",Value_ESP31T);                //ESP31T
fprintf(dataFile,"%3.5f,",Value_ESP32T);                //ESP32T
fprintf(dataFile,"%3.5f,",Value_SLOART);                //SLOART
fprintf(dataFile,"%3.5f,",Value_SPXCPT);                //SPXCPT
//
//---------------------------------------------------
//
// ** CONTINUOUS SATELLITE VARIABLES - NORMALIZED VALUES **
//
//-------------------------------------------------------
//
//
// Link 2 Communications Subsystem - Continuous Variables - Normalized Values
//
fprintf(dataFile,"%3.5f,",Norm_C_plus_150V);            //C+150V
fprintf(dataFile,"%3.5f,",Norm_C2A_plus_1V);            //C2A+1V
fprintf(dataFile,"%3.5f,",Norm_C2A_minus_1V);           //C2A-1V
fprintf(dataFile,"%3.5f,",Norm_C2A28V);                 //C2A28V
fprintf(dataFile,"%3.5f,",Norm_C2B_plus_1V);            //C2B+1V
fprintf(dataFile,"%3.5f,",Norm_C2B_minus_1V);           //C2B-1V
fprintf(dataFile,"%3.5f,",Norm_C2B28V);                 //C2B28V
fprintf(dataFile,"%3.5f,",Norm_CC_plus_50V);            //CC+150V
fprintf(dataFile,"%3.5f,",Norm_CC_plus_60V);            //CC+60V
fprintf(dataFile,"%3.5f,",Norm_CC_minus_23V);           //CC-23V
fprintf(dataFile,"%3.5f,",Norm_CC_minus_60V);           //CC-60V
fprintf(dataFile,"%3.5f,",Norm_CCINIT);                 //CCINIT
fprintf(dataFile,"%3.5f,",Norm_CCS1CV);                 //CCS1CV
fprintf(dataFile,"%3.5f,",Norm_CCS2CV);                 //CCS2CV
```

```
fprintf(dataFile,"%3.5f,",Norm_CCS3CV);            //CCS3CV
//
// Link 1 Communications Subsystem - Continuous Variables - Normalized Values
//
fprintf(dataFile,"%3.5f,",Norm_CCT1AV);            //CCT1AV
fprintf(dataFile,"%3.5f,",Norm_CCT1AW);            //CCT1AW
fprintf(dataFile,"%3.5f,",Norm_CCT1BV);            //CCT1BV
fprintf(dataFile,"%3.5f,",Norm_CCT1BW);            //CCT1BW
fprintf(dataFile,"%3.5f,",Norm_CECA5V);            //CECA5V
fprintf(dataFile,"%3.5f,",Norm_CECB5V);            //CECB5V
fprintf(dataFile,"%3.5f,",Norm_CPA1AT);            //CPA1AT
fprintf(dataFile,"%3.5f,",Norm_CPA1AV);            //CPA1AV
fprintf(dataFile,"%3.5f,",Norm_CPA1AW);            //CPA1AW
fprintf(dataFile,"%3.5f,",Norm_CPA1BT);            //CPA1BT
fprintf(dataFile,"%3.5f,",Norm_CPA1BV);            //CPA1BV
fprintf(dataFile,"%3.5f,",Norm_CPA1BW);            //CPA1BW
fprintf(dataFile,"%3.5f,",Norm_CT1AMT);            //CT1AMT
fprintf(dataFile,"%3.5f,",Norm_CT1BMT);            //CT1BMT
//
// Propulsion Subsystem - Continuous Variables - Raw Values
//
fprintf(dataFile,"%3.5f,",Norm_GGPRES);            //GGPRES
fprintf(dataFile,"%3.5f,",Norm_LVLSEL);            //LVLSEL
fprintf(dataFile,"%3.5f,",Norm_P_plus_XFBT);       //P+XFBT
fprintf(dataFile,"%3.5f,",Norm_P_plus_XPTP);       //P+XPTP
fprintf(dataFile,"%3.5f,",Norm_P_plus_XTOT);       //P+XTOT
fprintf(dataFile,"%3.5f,",Norm_P_plus_XVBT);       //P+XVBT
fprintf(dataFile,"%3.5f,",Norm_PFDIVT);            //PFDIVT
fprintf(dataFile,"%3.5f,",Norm_PGGFPT);            //PGGFPT
fprintf(dataFile,"%3.5f,",Norm_PHLT_plus_T);       //PHLT+T
fprintf(dataFile,"%3.5f,",Norm_PHLT_minus_T);      //PHLT-T
```

```
fprintf(dataFile,"%3.5f,",Norm_PHT_plus_HT);          //PHT+HT
fprintf(dataFile,"%3.5f,",Norm_PHT_minus_HT);         //PHT-HT
fprintf(dataFile,"%3.5f,",Norm_PPLN1P);               //PPLN1P
fprintf(dataFile,"%3.5f,",Norm_PPLN2P);               //PPLN2P
fprintf(dataFile,"%3.5f,",Norm_PVT_plus_HT);          //PVT+HT
fprintf(dataFile,"%3.5f,",Norm_PVT_minus_HT);         //PVT-HT
fprintf(dataFile,"%3.5f,",Norm_PVVT_plus_T);          //PVVT+T
fprintf(dataFile,"%3.5f,",Norm_PVVT_minus_T);         //PVVT-T
fprintf(dataFile,"%3.5f,",Norm_P_minus_XCLT);         //P-XCLT
fprintf(dataFile,"%3.5f,",Norm_P_minus_XFBT);         //P-XFBT
fprintf(dataFile,"%3.5f,",Norm_P_minus_XPTP);         //P-XPTP
fprintf(dataFile,"%3.5f,",Norm_P_minus_XTIT);         //P-XTIT
fprintf(dataFile,"%3.5f,",Norm_P_minus_XTOT);         //P-XTOT
fprintf(dataFile,"%3.5f,",Norm_P_minus_XVBT);         //P-XVBT
//
// Electrical Power Subsystem - Continuous Variables - Normalized Values
//
fprintf(dataFile,"%3.5f,",Norm_ECUSBI);               //ECUSBI
fprintf(dataFile,"%3.5f,",Norm_ECUSBV);               //ECUSBV
fprintf(dataFile,"%3.5f,",Norm_EED1AV);               //EED1AV
fprintf(dataFile,"%3.5f,",Norm_EED1BV);               //EED1BV
fprintf(dataFile,"%3.5f,",Norm_EED5AV);               //EED5AV
fprintf(dataFile,"%3.5f,",Norm_EED5BV);               //EED5BV
fprintf(dataFile,"%3.5f,",Norm_EPBA1T);               //EPBA1T
fprintf(dataFile,"%3.5f,",Norm_EPBA2T);               //EPBA2T
fprintf(dataFile,"%3.5f,",Norm_EPBB1T);               //EPPB1T
fprintf(dataFile,"%3.5f,",Norm_EPBB2T);               //EPBB2T
fprintf(dataFile,"%3.5f,",Norm_EPBC1T);               //EPBC1T
fprintf(dataFile,"%3.5f,",Norm_EPBC2T);               //EPBC2T
fprintf(dataFile,"%3.5f,",Norm_EPBSAI);               //EPBSAI
fprintf(dataFile,"%3.5f,",Norm_EPLP2T);               //EPLP2T
```

```
fprintf(dataFile,"%3.5f,",Norm_EPSBAI);          //EPSBAI
fprintf(dataFile,"%3.5f,",Norm_EPSBAV);          //EPSBAV
fprintf(dataFile,"%3.5f,",Norm_EPSBBI);          //EPSBBI
fprintf(dataFile,"%3.5f,",Norm_EPSBBV);          //EPSBBV
fprintf(dataFile,"%3.5f,",Norm_EPSBCI);          //EPSBCI
fprintf(dataFile,"%3.5f,",Norm_EPSBCV);          //EPSBCV
fprintf(dataFile,"%3.5f,",Norm_EPSDBV);          //EPSDBV
fprintf(dataFile,"%3.5f,",Norm_EPSEBT);          //EPSEBT
fprintf(dataFile,"%3.5f,",Norm_EPSEDT);          //EPSEDT
fprintf(dataFile,"%3.5f,",Norm_EPSEET);          //EPSEET
fprintf(dataFile,"%3.5f,",Norm_EPSLBI);          //EPSLBI
fprintf(dataFile,"%3.5f,",Norm_EPSPBV);          //EPSPBV
fprintf(dataFile,"%3.5f,",Norm_EPUC1T);          //EPUC1T
fprintf(dataFile,"%3.5f,",Norm_EPUC2T);          //EPUC2T
fprintf(dataFile,"%3.5f,",Norm_EPUN1T);          //EPUN1T
fprintf(dataFile,"%3.5f,",Norm_ESCLBI);          //ESCLBI
fprintf(dataFile,"%3.5f,",Norm_ESP31T);          //ESP31T
fprintf(dataFile,"%3.5f,",Norm_ESP32T);          //ESP32T
fprintf(dataFile,"%3.5f,",Norm_SLOART);          //SLOART
fprintf(dataFile,"%3.5f,",Norm_SPXCPT);          //SPXCPT
//
//-------------------------------------------------
//
// ** FLAGS ON CONTINUOUS SATELLITE VARIABLES **
//
//-------------------------------------------------
//
// Link 2 Communications Subsystem - FLAGS
fprintf(dataFile,"%d,",Flag_C_plus_150V);        //C+150V
fprintf(dataFile,"%d,",Flag_C2A_plus_1V);        //C2A+1V
fprintf(dataFile,"%d,",Flag_C2A_minus_1V);       //C2A-1V
```

```
fprintf(dataFile,"%d,",Flag_C2A28V);                //C2A28V
fprintf(dataFile,"%d,",Flag_C2B_plus_1V);           //C2B+1V
fprintf(dataFile,"%d,",Flag_C2B_minus_1V);          //C2B-1V
fprintf(dataFile,"%d,",Flag_C2B28V);                //C2B28V
fprintf(dataFile,"%3.3f\t",Flag_CC_plus_50V);       //CC+50V
fprintf(dataFile,"%3.3f\t",Flag_CC_plus_60V);       //CC+60V
fprintf(dataFile,"%3.3f\t",Flag_CC_minus_23V);      //CC-23V
fprintf(dataFile,"%d,",Flag_CC_minus_60V);          //CC-60V
fprintf(dataFile,"%d,",Flag_CCINIT);                //CCINIT
fprintf(dataFile,"%d,",Flag_CCS1CV);                //CCS1CV
fprintf(dataFile,"%d,",Flag_CCS2CV);                //CCS2CV
fprintf(dataFile,"%d,",Flag_CCS3CV);                //CCS3CV
//
// Link 1 Communications Subsystem - FLAGS
//
fprintf(dataFile,"%d,",Flag_CCT1AV);                //CCT1AV
fprintf(dataFile,"%d,",Flag_CCT1AW);                //CCT1AW
fprintf(dataFile,"%d,",Flag_CCT1BV);                //CCT1BV
fprintf(dataFile,"%d,",Flag_CCT1BW);                //CCT1BW
fprintf(dataFile,"%d,",Flag_CECA5V);                //CECA5V
fprintf(dataFile,"%d,",Flag_CECB5V);                //CECB5V
fprintf(dataFile,"%d,",Flag_CPA1AT);                //CPA1AT
fprintf(dataFile,"%d,",Flag_CPA1AV);                //CPA1AV
fprintf(dataFile,"%d,",Flag_CPA1AW);                //CPA1AW
fprintf(dataFile,"%d,",Flag_CPA1BT);                //CPA1BT
fprintf(dataFile,"%d,",Flag_CPA1BV);                //CPA1BV
fprintf(dataFile,"%d,",Flag_CPA1BW);                //CPA1BW
fprintf(dataFile,"%d,",Flag_CT1AMT);                //CT1AMT
fprintf(dataFile,"%d,",Flag_CT1BMT);                //CT1BMT
//
// Propulsion Subsystem - FLAGS
```

```
//
fprintf(dataFile,"%d,",Flag_GGPRES);            //GGPRES
fprintf(dataFile,"%d,",Flag_LVLSEL);            //LVLSEL
fprintf(dataFile,"%d,",Flag_P_plus_XFBT);       //P+XFBT
fprintf(dataFile,"%d,",Flag_P_plus_XPTP);       //P+XPTP
fprintf(dataFile,"%d,",Flag_P_plus_XTOT);       //P+XTOT
fprintf(dataFile,"%d,",Flag_P_plus_XVBT);       //P+XVBT
fprintf(dataFile,"%d,",Flag_PFDIVT);            //PFDVIT
fprintf(dataFile,"%d,",Flag_PGGFPT);            //PGGFPT
fprintf(dataFile,"%d,",Flag_PHLT_plus_T);       //PHLT+T
fprintf(dataFile,"%d,",Flag_PHLT_minus_T);      //PHLT-T
fprintf(dataFile,"%d,",Flag_PHT_plus_HT);       //PHT+HT
fprintf(dataFile,"%d,",Flag_PHT_minus_HT);      //PHT-HT
fprintf(dataFile,"%d,",Flag_PPLN1P);            //PPLN1P
fprintf(dataFile,"%d,",Flag_PPLN2P);            //PPLN2P
fprintf(dataFile,"%d,",Flag_PVT_plus_HT);       //PVT+HT
fprintf(dataFile,"%d,",Flag_PVT_minus_HT);      //PVT-HT
fprintf(dataFile,"%d,",Flag_PVVT_plus_T);       //PVVT+T
fprintf(dataFile,"%d,",Flag_PVVT_minus_T);      //PVVT-T
fprintf(dataFile,"%d,",Flag_P_minus_XCLT);      //P-XCLT
fprintf(dataFile,"%d,",Flag_P_minus_XFBT);      //P-XFBT
fprintf(dataFile,"%d,",Flag_P_minus_XPTP);      //P-XPTP
fprintf(dataFile,"%d,",Flag_P_minus_XTIT);      //P-XTIT
fprintf(dataFile,"%d,",Flag_P_minus_XTOT);      //P-XTOT
fprintf(dataFile,"%d,",Flag_P_minus_XVBT);      //P-XVBT
//
// Electrical Power Subsystem - FLAGS
//
fprintf(dataFile,"%d,",Flag_ECUSBI);            //ECUSBI
fprintf(dataFile,"%d,",Flag_ECUSBV);            //ECUSBV
fprintf(dataFile,"%d,",Flag_EED1AV);            //EED1AV
```

```
fprintf(dataFile,"%d,",Flag_EED1BV);        //EED1BV
fprintf(dataFile,"%d,",Flag_EED5AV);        //EED5AV
fprintf(dataFile,"%d,",Flag_EED5BV);        //EED5BV
fprintf(dataFile,"%d,",Flag_EPBA1T);        //EPBA1T
fprintf(dataFile,"%d,",Flag_EPBA2T);        //EPBA2T
fprintf(dataFile,"%d,",Flag_EPBB1T);        //EPBB1T
fprintf(dataFile,"%d,",Flag_EPBB2T);        //EPBB2T
fprintf(dataFile,"%d,",Flag_EPBC1T);        //EPBC1T
fprintf(dataFile,"%d,",Flag_EPBC2T);        //EPBC2T
fprintf(dataFile,"%d,",Flag_EPBSAI);        //EPBSAI
fprintf(dataFile,"%d,",Flag_EPLP2T);        //EPLP2T
fprintf(dataFile,"%d,",Flag_EPSBAI);        //EPSBAI
fprintf(dataFile,"%d,",Flag_EPSBAV);        //EPSBAV
fprintf(dataFile,"%d,",Flag_EPSBBI);        //EPSBBI
fprintf(dataFile,"%d,",Flag_EPSBBV);        //EPSBBV
fprintf(dataFile,"%d,",Flag_EPSBCI);        //EPSBCI
fprintf(dataFile,"%d,",Flag_EPSBCV);        //EPSBCV
fprintf(dataFile,"%d,",Flag_EPSDBV);        //EPSDBV
fprintf(dataFile,"%d,",Flag_EPSEBT);        //EPSEBT
fprintf(dataFile,"%d,",Flag_EPSEDT);        //EPSEDT
fprintf(dataFile,"%d,",Flag_EPSEET);        //EPSEET
fprintf(dataFile,"%d,",Flag_EPSLBI);        //EPSLBI
fprintf(dataFile,"%d,",Flag_EPSPBV);        //EPSPBV
fprintf(dataFile,"%d,",Flag_EPUC1T);        //EPUC1T
fprintf(dataFile,"%d,",Flag_EPUC2T);        //EPUC2T
fprintf(dataFile,"%d,",Flag_EPUN1T);        //EPUN1T
fprintf(dataFile,"%d,",Flag_ESCLBI);        //ESCLBI
fprintf(dataFile,"%d,",Flag_ESP31T);        //EPS31T
fprintf(dataFile,"%d,",Flag_ESP32T);        //EPS32T
fprintf(dataFile,"%d,",Flag_SLOART);        //SLOART
fprintf(dataFile,"%d,",Flag_SPXCPT);        //SPXCPT
```

```
//
//---------------------------------------------------
//
// ** DISCRETES **
//
//---------------------------------------------------
//
// Link 2 Communications Subsystem - Discretes
//
fprintf(dataFile,"%d,",State_C2ASPB);          //C2ASPB
fprintf(dataFile,"%d,",State_EDABRB);          //EDABRB
fprintf(dataFile,"%d,",State_EDBBRB);          //EDBBRB
fprintf(dataFile,"%d,",State_EDTUBB);          //EDTUBB
fprintf(dataFile,"%d,",State_EDTUAB);          //EDTUAB
fprintf(dataFile,"%d,",State_EDTUDB);          //EDTUDB
fprintf(dataFile,"%d,",State_EKG2AB);          //EKG2AB
fprintf(dataFile,"%d,",State_EKG2BB);          //EKG2BB
fprintf(dataFile,"%d,",State_ERBUDB);          //ERBUDB
fprintf(dataFile,"%d,",State_ET2AAB);          //ET2AAB
fprintf(dataFile,"%d,",State_ET2AEB);          //ET2AEB
fprintf(dataFile,"%d,",State_ET2AOB);          //ET2AOB
fprintf(dataFile,"%d,",State_ET2BAB);          //ET2BAB
fprintf(dataFile,"%d,",State_ET2BEB);          //ET2BEB
fprintf(dataFile,"%d,",State_ET2BOB);          //ET2BOB
//
// Link 1 Communications Subsystem - Discretes
//
fprintf(dataFile,"%d,",State_CDCAMB);          //CDCAMB
fprintf(dataFile,"%d,",State_CDCBMB);          //CDCBMB
fprintf(dataFile,"%d,",State_CSW1PB);          //CSW1PB
fprintf(dataFile,"%d,",State_CSW2PB);          //CSW2PB
```

```
fprintf(dataFile,"%d,",State_CSW3PB);          //CSW3PB
fprintf(dataFile,"%d,",State_CSW4PB);          //CSW4PB
fprintf(dataFile,"%d,",State_CSW5PB);          //CSW5PB
fprintf(dataFile,"%d,",State_E1ASOB);          //E1ASOB
fprintf(dataFile,"%d,",State_E1BSOB);          //E1BSOB
fprintf(dataFile,"%d,",State_EDC1AB);          //EDC1AB
fprintf(dataFile,"%d,",State_EDECAB);          //EDECAB
fprintf(dataFile,"%d,",State_EDECBB);          //EDECBB
fprintf(dataFile,"%d,",State_EKG1AB);          //EKG1AB
fprintf(dataFile,"%d,",State_EKG1BB);          //EKG1BB
fprintf(dataFile,"%d,",State_EP1AHB);          //EP1AHB
fprintf(dataFile,"%d,",State_EP1ASB);          //EP1ASB
fprintf(dataFile,"%d,",State_EP1BSB);          //EP1BSB
fprintf(dataFile,"%d,",State_EP1BHB);          //EP1BHB
fprintf(dataFile,"%d,",State_ET1AAB);          //ET1AAB
fprintf(dataFile,"%d,",State_ET1AEB);          //ET1AEB
fprintf(dataFile,"%d,",State_ET1AOB);          //ET1AOB
fprintf(dataFile,"%d,",State_ET1BOB);          //ET1BOB
fprintf(dataFile,"%d,",State_ET1BAB);          //ET1BAB
fprintf(dataFile,"%d,",State_ET1BEB);          //ET1BEB
//
// Propulsion Subsystem - Discretes
//
fprintf(dataFile,"%d,",State_ACJEAB);          //ACJEAB
fprintf(dataFile,"%d,",State_ACJEBB);          //ACJEBB
fprintf(dataFile,"%d,",State_ACTEAB);          //ACTEAB
fprintf(dataFile,"%d,",State_ACTEBB);          //ACTEBB
fprintf(dataFile,"%d,",State_AGGAEB);          //AGGAEB
fprintf(dataFile,"%d,",State_AGGBEB);          //AGGBEB
fprintf(dataFile,"%d,",State_ASJEAB);          //ASJEAB
fprintf(dataFile,"%d,",State_ASJEBB);          //ASJEBB
```

```
fprintf(dataFile,"%d,",State_AVVEAB);          //AVVEAB
fprintf(dataFile,"%d,",State_AVVEBB);          //AVVEBB
//
// Electrical Power Subsystem - Discretes
//
fprintf(dataFile,"%d,",State_EAACMB);          //EAACMB
fprintf(dataFile,"%d,",State_EADBYB);          //EADBYB
fprintf(dataFile,"%d,",State_EASBPB);          //EASBPB
fprintf(dataFile,"%d,",State_EASBRB);          //EASBRB
fprintf(dataFile,"%d,",State_EBACMB);          //EBACMB
fprintf(dataFile,"%d,",State_EBAHEB);          //EBAHEB
fprintf(dataFile,"%d,",State_EBAK1B);          //EBAK1B
fprintf(dataFile,"%d,",State_EBAK2B);          //EBAK2B
fprintf(dataFile,"%d,",State_EBAK3B);          //EBAK3B
fprintf(dataFile,"%d,",State_EBARDB);          //EBARDB
fprintf(dataFile,"%d,",State_EBARTB);          //EBARTB
fprintf(dataFile,"%d,",State_EBBHEB);          //EBBHEB
fprintf(dataFile,"%d,",State_EBBK1B);          //EBBK1B
fprintf(dataFile,"%d,",State_EBBK2B);          //EBBK2B
fprintf(dataFile,"%d,",State_EBBK3B);          //EBBK3B
fprintf(dataFile,"%d,",State_EBBRDB);          //EBBRDB
fprintf(dataFile,"%d,",State_EBBRTB);          //EBBRTB
fprintf(dataFile,"%d,",State_EBCHEB);          //EBCHEB
fprintf(dataFile,"%d,",State_EBCK1B);          //EBCK1B
fprintf(dataFile,"%d,",State_EBCK2B);          //EBCK2B
fprintf(dataFile,"%d,",State_EBCK3B);          //EBCK3B
fprintf(dataFile,"%d,",State_EBCRDB);          //EBCRDB
fprintf(dataFile,"%d,",State_EBCRTB);          //EBCRTB
fprintf(dataFile,"%d,",State_EBDBYB);          //EBDBYB
fprintf(dataFile,"%d,",State_EBVLSB);          //EBVLSB
fprintf(dataFile,"%d,",State_ECACMB);          //ECACMB
```

```
fprintf(dataFile,"%d,",State_ECDBYB);            //ECDBYB
fprintf(dataFile,"%d,",State_EDUVSB);            //EDUVSB
fprintf(dataFile,"%d,",State_ESP1PB);            //ESP1PB
fprintf(dataFile,"%d,",State_ESP1RB);            //ESP1RB
fprintf(dataFile,"%d,",State_ESP2PB);            //ESP2PB
fprintf(dataFile,"%d,",State_ESP2RB);            //ESP2RB
fprintf(dataFile,"%d,",State_ESP3PB);            //ESP3PB
fprintf(dataFile,"%d,",State_ESP3RB);            //ESP3RB
fprintf(dataFile,"%d,",State_ESP4PB);            //ESP4PB
fprintf(dataFile,"%d,",State_ESP4RB);            //ESP4RB
// ***********************************
// ** End of simulated satellite data **
// ***********************************
//
//
// **************************************************
// ** Begin Satellite Controller Performance Measures **
// **************************************************
//
fprintf(dataFile,"%d,",dcCursorX);               //cursor coordinate
fprintf(dataFile,"%d\n",dcCursorY);              //cursor coordinate
//
//      Ncte the existence of the '\n' in the format statement follwing
//      following the last 'fprint in this list'. For appropriate formatting, it
//      is necesary for the last fprint to have a '\n' as
//      shown. If additional variables are added to the list,
//      maintain the standard of having the last collected
//      variable, insert a '\n' as part of the format.
// Button pressed (included mouse click and actions caused by voice command)- TBD
// Voice Recognition Beginning Point (a flag?) - TBD
// Phrase Recognized (Can we capture the phrase that the speech recognizer
```

// decided :he controller said?) - TBD

DATA FILES

The data file created during a session is a comma delimited text file. This format was selected primarily to facilitate importing the data into commercial spreadsheets (e.g., Excel) or into commercial statistical packages (e.g., SAS) for post—experiment analysis. The text format also aids the experimenter is taking a "quick look" at the data. The data files can be opened and viewed using standard word processing packages. The drawback to this format is that the data files are larger than with some other formats, such as binary.

Each row of data file contains a value for each variable in the data collection list. The order of the variables in each row is the same as the order of the variables in the data collection list. For example, if the data collection list is:

```
fprintf(dataFile,"%d,",dcMode);             // start/stop/pause/resume
fprintf(dataFile,"%d,",dcEventCount);           // operator entered
//fprintf(dataFile,"%d,",dcAnomalyFlag);    // operator entered
fprintf(dataFile,"%d,",dcAnomalyType);      // operator entered
```

Then the first column of each row would contain *dcMode*, the second column *dcEventCount* , and the third column *dcAnomalyType*. *dcAnomalyFlag* would not be collected as it is "commented out" in the data collection list.

Each row represents a different point in time. The amount of time between each row is selected by the experimenter from the DATA COLLECTION field in the experimenter's console.

Appendix 7 – Emerging Human-System Interface Technologies

#	Technology	Reference	Description
1	Speech Recognition	***Joint C2 Battlelab TAPTalk Project*** Personal conversations with David Williamson (AFRL-Wright Patterson), a demonstration video, and several unpublished reports, and the following published report: Williamson, D.T., and Barry, T.P. "The Design and Evaluation of a Speech Interface for Generation of Air Tasking Orders." *Proceedings of the Human Factors and Ergonomic Society*, 2000.	**Excerpts** This paper discusses the design, implementation, and evaluation of a prototype speech recognition interface to the Theater Air Planning (TAP) module of Theater Battle Management Core Systems (TBMCS). This effort was in support of a Kenney Battlelab Initiative proposal submitted to the Command and Control Battlelab at Hurlburt Field, FL to assess the operational benefits of speech recognition for data entry applications in a Joint Air Operations Center environment. Several factors contributing to the design of the "TAPTalk" speech interface included interviews with subject matter experts, speech system selection, grammar development, and integration into TAP, which required only minor modification of existing software. Results from the two week operational assessment with sixteen subjects from the Command and Control Training and Innovation Group, numbered Air Forces, Navy, and Marine Corp indicated that the Theater Air Planning process could be accomplished significantly faster with no increase in error rates. Subjectively, the sixteen planners unanimously agreed that the TAPTalk speech interface was a valuable addition to TAP and would recommend its inclusion in a future upgrade. Recommendations for further improving the TAPTalk system are discussed. The primary objective was to determine if speech recognition had reached a level of maturity for it to be considered a viable input technology for C2 software applications. Three areas were investigated: 1. Menu navigation – goal was to provide a short command that would take the operator directly to the desired window (rather than requiring a command through each level of the menu). 2. Data entry – using data fields to minimize input requirements.

#	Technology	Reference	Description
			3. Database query – to gain rapid access to information. Implementation: The speech implementation included the following components: • Nuance 6, a speaker independent COTS speech recognition package from Nuance Communications • A vocabulary of 476 unique words to implement the TAPTalk system • These words were contained in 16 primary grammars and 69 sub-grammars • An Andrea Electronics ANC-500 active noise cancellation monaural headsets with mute switches to minimize interference from adjacent subjects and environmental noise. Results: The operational assessment showed that operators were quick to adapt to voice communication with the TBMCS application. When the appropriate vocabulary was used, the prototype speech recognition software recognized the subject matter experts' (SMEs) verbal communication 97% of the time. The assessment concluded that speech recognition technology was mature enough to act as a C2 computer interface. Speech recognition proved extremely useful for manipulating menus and locating infrequently used information (i.e. the items that were hard to remember and difficult to locate). The operational assessment also showed that speech recognition can reduce the time

#	Technology	Reference	Description
			required for ATO development. The data collected, even though limited, supported the premise that the training of new TBMCS operators should be reduced by this technology. Though no extended endurance assessment sessions were conducted, SMEs felt that the use of speech recognition will reduce operator fatigue. An Analysis of Variance for task completion times during this formal evaluation phase revealed a significant advantage of the TAPTalk system over the conventional mouse and keyboard input method ($F=14.99$, $p < .008$). <u>Recommendations:</u> This initiative determined that the current state of the art for speech recognition is advanced enough to enhance software applications. The Air Force needs to take advantage of speech technology and field speech recognition for the warfighter as soon as possible. To accomplish this, in accordance with the requirements established is the Capstone Requirements Document and applicable Mission Needs Statements, the AC2ISRC must identify those C2ISR applications and processes that can benefit from the integration of speech recognition capabilities. In addition, the AC2ISRC must develop a transition plan to begin the process of putting this capability into the field. To begin this process we recommend two initial steps. The first step would be to develop a joint concept of operations and vision through a Federated Battlelab Initiative for FY 00. This would result in a joint standard for speech to ensure interoperability among the services. Our second step would introduce speech recognition into the Defense Information Infrastructure Common Operation Environment (DII COE) for general use. Implementation of these tools in the daily workplace is required to improve the proficiency, make data entry easier, accelerate the learning curve, reduce the workload of the operator, and provide an enabler for the integration of speech recognition.

#	Technology	Reference	Description
			<u>Update:</u> Based upon a conversation with David Williamson, the next phase of his work will focus on making use of the OOA Architecture to integrate other input and output modalities with speech to provide a multimodal user interface capability.
2	Multimodal Interfaces	***SPAWAR Multimodal Watchstation (MMWS)*** Based upon direct contact with the MMWS researchers, and multiple demonstrations, including participation is a group training session Several reports on operational field testing and usability tests such as the one report in the following reference: Osga, G., Nugent, B., and Campbell, N. "Impact of Control/Display Configuration and Graphic User Interface Methods on Performance Speed and Accuracy," *SPAWAR Systems Center publication.*	**Analysis** Currently, the Navy SPAWAR Multimodal Watchstation (MMWS), remains the most appropriate "pathfinder" project for the SOC advanced multimodal interface research and development project. In addition to being the most ambitious and mature project identified to date, it is the most representative of the current SOC project with respect to objectives and approach as seen in the following points: • The main objective is to exploit the ability of advanced HSI and intelligent software technologies to create a workstation that will significantly enhance the performance of the 21st century • MMWS is a four-year project sponsored by ONR with technology feeds from 12 years of ONR research (e.g., TADMUS, Speech Sentinel, 3D Audio, etc.) • The approach called for creation of a test bed to determine performance benefits the advanced technologies over current tactical watchstations (e.g. Aegis operator console). • The test bed hardware and software are mainly COTS/GOTS components conducive to flexibility and evolvability and will run on open systems architecture and software environment. • The initial test bed comprised of multi-modal control

#	Technology	Reference	Description
			and input methods including touch, speech, and natural language, combined with multiple flat-panel displays, 3D audio, and advanced information management technologies • Test bed incorporates built-in real-time operator performance and workload monitoring and recording capabilities • A user-centered design approach with extensive up-front task analysis and formative usability evaluation with design iteration • Currently in sixth iteration with three years of usability testing involving 110 subjects representative of the potential user population • Next test bed upgrade is combined flat-panel and partially immersive head-mounted visual displays which present head-tracked visualization With three years of usability testing and design iteration, the MMWS will definitely serve as a pathfinder project with respect to many aspects of future SOC testbed design and development. Direct contact with the MMWS principal investigators has provided an enormous amount of detailed information on all aspects of design, methodology, research findings, lessons learned etc., The degree to which this information can be directly applied to our project to expedite implementation and reduce various risks is largely dependent upon the extent to which the MMWS application environment correlates with the SOC application domain. A brief description the MMWS application environment and some preliminary comparisons with the SOC environment are presented below. First, it should be pointed out, that with the possible exception of the

#	Technology	Reference	Description
			Joint Air Operations Center study reported earlier, no other advanced multimodal research project identified provides as many parallels with the space command center applications environment in terms of operational setting and general task requirements as does the MMWS project. The MMWS application environment can be summarized as follows: • At the highest level the MMWS is a next-generation military command, control and communications (C3) system. If it is actually deployed it will become an integral part of the overall Joint Forces C4I framework mandated by Vision 2010. The same could be said of the SOC. • At the next level down, it is currently being characterized as a "multi-mission warfare coordination system." It is not a single-mission system; it must be designed with the flexibility to support a variety of warfare coordination missions. Similarly, the SOC should be designed with other space operations missions in mind. • The MMWS will be used in various operational settings including ships (i.e., scheduled to go aboard the Surface Combatant 21st Century land attack destroyer), aircraft, and ground command and control facilities. The main operational setting for the SOC is likely to be space operations command and control ground stations. • Evaluations to-date have focused on accessing the MMWS's advanced HSI technologies under a demanding warfare coordination mission in the form of a tactical air defense application scenarios. It is at the last item that raises questions as to the applicability of the MMWS research findings. There would appear to be major differences between a tactical air defense application environment and a space operations application environment. For example, the demanding littoral tactical air warfare application used for most of the MMWS research can be characterized as a fast-paced stressful environment with a high

#	Technology	Reference	Description
			degree of message and voice communications between personnel within the command center and personnel at various remote locations to clarify the tactical situation. The demands of certain space operations such as a launch or a critical spacecraft maneuver may result in similar stressful high-paced conditions. The MMWS evaluation applications would appear to have some relevance to the space operations as the general task level. For example, both application domains require continuous monitoring of the health and safety status of sub-systems for high-value military assets. Both applications require operators to aggregate information from multiple information sources, access database information, input data and parameters, input commands, evaluate trends, problem solving to resolve anomalies, communicate with other operations personnel, and timely and accurate decisions, The Space and Naval Warfare System Center has conducting usability research on the Multimodal Watchstation (MMWS). They have performed a series of experiments examining various control and display configurations including pull-down menus, off-screen function keys, on-screen function keys, voice and trackball. Based upon operator speed and accuracy and preference data it was found that design options using touch screen, voice entry, and touch entry function arrays were among the fastest methods. Pull down menus, as found on most commercial software products, were among the slowest function activation methods tested. Combinations of voice and touch activation with function key activation by alternate hand distribute workload between hands and voice supporting fast performance. Update: Based upon a recent discussion with Dr. Glen Osga, Program Manager, current research and development is concentrated on the improving the effectiveness of the decision aids and workload management aspect of MMWS design. The MMWS software is currently being ported to run on the Lockheed Martin Eagan Valiant

#	Technology	Reference	Description
			Console.
3	Multipe HSI Techrologies	***Command Post of the Future***, DARPA Information Systems Office presentation. Page, W. (1999).	**Excerpts** The commander's job, in the future as in the past, will be to make decisions and monitor their execution in the midst of great uncertainty. He will need to assess the changing situation, select the best course of action, and monitor its execution. In the future, his success will depend on using information dominance to increase the speed and precision of those decisions. At the same time, survival will depend on being small and mobile. Large command complexes will not survive in the highly lethal, future battlefield. The commander's portal into this information environment will need to be easily operated by a small, distributed staff. The goal of CPOF is to shorten the commander's decision cycle to stay ahead of the adversary's ability to react. To achieve this operational goal, the technical objective is to develop the technology necessary to create an adaptive, decision-centered, information visualization environment for the future commander and his immediate staff. Current technology is flooding the commander with messages, images, and data which require increasing numbers of people and computers to process, interpret, integrate, and understand the incoming information streams. The CPoF system will provide the commander information about the battlespace in a form that will enhance his cognitive processes. Decreasing the uncertainties, unknowns, and the fragmented pictures of the battlefield, while enhancing the Commander's ability to make decisions and direct their execution in an environment of great uncertainty. In building and maintaining situation awareness the system must avoid simply increasing the quantity of data provided to the commander. The system will provide information by exception rather than as the norm in a graphical form, where appropriate, to assist the Commander in finding critical vulnerabilities, project trends, and development of decision centered solutions. In order to facilitate the Commander's visualization of the battlefield we will develop / integrate

#	Technology	Reference	Description
			advanced concepts for a Command Post which will exploit recent advancements in human computer interaction technologies incorporating interactive 3D visualization, interactive 3D techniques, uncertainty presentation, temporal presentation, 3d symbology, Natural Language (NL) processing, and Knowledge Base (KB) querying technologies. This system will additionally provide for collaborative planning using on-screen teleconferencing, shared map planning and electronic white boards. <u>Project Objectives:</u> Provide the commander and his staff with an environment that will expand their cognitive processes, while enhancing their ability to make decisions and direct their execution. The system will provide a means to rapidly visualize, interpret, integrate and analyze information about the battlespace, while decreasing the uncertainties, unknowns, and the fragmented pictures of the battlespace. In order to facilitate this visualization we will develop / integrate advanced concepts in Human Computer Interaction (HCI) technologies, interactive 3D visualization techniques, uncertainty presentation, temporal presentation, Natural Language (NL) processing, and Knowledge Base (KB) querying technologies, collaborative planning, teleconferencing, shared map planning and electronic white boards. The development and integration of these technologies will be in conjunction with operational units: USMC -Special Purpose Marine Air Ground Task Force (Experimental) a MEU size force of 3,500 and the US Army 525th MI Brigade (Airborne) force size of 1,100 supporting 18th Airborne Corps with force size of 70,000,US Navy - Extending the Littoral Battlespace (ELB) ACTD. <u>Pay Offs:</u> This program will provide the operational commander and his immediate staff in the command center with improved capabilities to perform their crucial functions. In particular, they will be able to: More rapidly recognize, understand and explore the implications of changes in the battlespace. Focus their experience, judgment, and training to allow

#	Technology	Reference	Description
			"naturalistic decision making" that directly links perceptions of the battlespace, decision making, and battle management. Develop and communicate the commander's estimate and concept of the operation both throughout the command center itself and to those virtually present in the center. Project the commander's virtual presence into the battlespace to enhance communication and comprehension. Increase the speed of the C3I cycle, particularly those activities centered in the command center itself. Overall, the payoffs will be measurable improvement in the speed and quality of C3I. <u>Challenges:</u> The adaptation to changing contexts (operating environments, missions, etc.), organizational adaptation (virtual organizations), as well as adaptation to different users and modes of use by the same user. Protection against information warfare. Assessment and evaluation to include success in defeating these adversary operations. Develop CPoF to support joint operations while relying on many legacy systems, which at the JTF component level, are service specific. Develop the ability of the CPoF to work in both well-organized coalitions (e.g. NATO and Korea) as well as less well-structured ad hoc coalitions. Improved C2 while dealing with potential operational missions: better battlespace visualization, information enabled organizations, adaptive decision making, agile battle management, significant increases in force effectiveness and efficiency. Trade-offs between the need for C2 processing speed, completeness of information, robustness of system, currency of information, capacity to project future situations, agility and quality of decision making, and battle management will be crucial. <u>Conduct User Experiments</u>: Conduct a continuous series of experiments with operational users to determine the human-factors and information requirements for the CPoF and to evaluate the best mix of technologies to assist the user in achieving and maintaining accurate and timely battlespace understanding, during simulated exercises. The potential

#	Technology	Reference	Description
			mix of technologies will include: • high resolution displays • electronic sand tables, • interactive 3D visualization and exploration techniques • speech and gesture interaction • natural language queries • collaboration support • decision-centered information management technologies. <u>Build and Demonstrate an Initial CPOF Prototype</u>: Develop an initial CPOF capability using near-term technology, to include large screen projection displays, non-stereoscopic table displays, simple speech commands, battlefield visualization with semantic zooming and 2-D graphics synthesis, and information integration. Demonstrate the prototype in Marine and Navy exercises. <u>Build and Demonstrate Final CPOF Prototype</u>: Enhance the baseline system by developing and incorporating advanced technology, such as interactive 3D visualization, stereoscopic displays, full natural language understanding and knowledge-based query capability, and decision-centered information management. Demonstrate the prototype system in Navy, Marine, and Army exercises. <u>Project Status</u>: Based upon the current experimentation schedule we expected to see results regarding the effectiveness of the new HSI technologies before the end of 1999. Experiments with a Pilot Study of "Tailored Visualizations for Building Situation Awareness" was scheduled to begin in September, 1999, and a Pilot Study of Tailored Visualizations to Support Single COA Comprehension" schedule for October, 1999. Numerous attempts to obtain updates from the Program Manager were unsuccessful. As of April 2002 there still appears to be groups working under CPoF funding, but there efforts seem to be fragmented.

#	Technology	Reference	Description
4	Multiple HSI Technologies	***AFRL UCAV Project*** "Uninhabited Combat Air Vehicle Controls and Displays for Suppression of Enemy Air Defense", *CSERIAC Gateway Publication* Volume XI number 1 (2000)	**Excerpts** Uninhabited is used to distinguish the new aircraft, enabled by new technologies, from those now in operation. UCAV's will be new, high-performance aircraft that are more effective for particular missions than are their inhabited counterparts. The *Human Effectiveness Directorate* of the AFRL is conducting a UCAV Operator Vehicle Interface (OVI) program to research issues with the human operator control stations for the UCAV. There are two program objectives. First, quantify UCAV control station requirements for the 21015 SEAD mission to evaluate automatic versus manual function tradeoffs that will enable a single operator to manage multiple UCAVs. Second, design operator-vehicle interfaces that integrate control/display technologies and decision-aiding features so that the system (the operator plus the UCAVs) can successfully accomplish all mission requirements. Since the system exists only as a concept, the research uses a simulated system. The UCAV operator's console will be highly automated, and there are critical human factors issues concerning the operator's interaction with that automation. The operator will be responsible for establishing system goals, monitoring and directing automated subsystems, and ensuring the overall

#	Technology	Reference	Description
			success of the mission. However, experience has shown that automation can have both desirable and undesirable effects. While automation can greatly improve the performance by taking over tasks that are performed poorly by a human operator or by reducing operator workload, high levels of automation cause the operator to become a system monitor, a task humans do poorly when they are not in the "decision-making loop." In fact, the term "clumsy automation" is often used to describe automation that is inconsistent or incompatible with the way humans think. With clumsy automation, there often is little or no feedback to the operator regarding system intent or performance. As a result, operators can be surprised by the behavior of an automatic system, which often leads to unanticipated – and sometimes undesirable – outcomes. The UCAV OVI program is performing analyses, design/redesign, and evaluation to develop a set of design guidelines for applying automation and human-computer interface (HCI) technologies. With a near-infinite number of design possibilities, we are using subject-matter experts—former Air Force pilots who have flown SEAD missions to decompose the mission to develop a design requirements scenario (very similar to a concept of operations). This identifies functional and information requirements for the control station design. The requirements, analyses, and decompositions then serve as the basis for developing

#	Technology	Reference	Description
			conceptual OV I designs and for evaluating their usefulness for multiple UCAV control within the SEAD context. Our first prototype OVI control station consisted of three 20-inch (diagonal measurement) liquid crystal displays (LCDs) placed side-by-side in a wraparound console, occupying about 100 degrees of the operator's field-of-view. During the evaluation, a computer mouse and keyboard voice recognition interface was also demonstrated to the participants after the formal data sessions for subjective impressions and critique.
5	Multipb HSI Technologies	***DARPA Communicator Program***	**Overview** The goal of the Defense Advanced Research Projects Agency (DARPA) Communicator program is to advance the state of the art of conversational human computer interfaces, especially for mobile environments, such as battlefields, where interface requirements, system robustness, and spoken dialogue strategies present unique and challenging demands for the interface developer. Current commercial interfaces support simple spoken commands and replies, in a limited domain, with strict human-computer interaction rules. The Communicator program goal is to support more robust and complex conversational interaction, where both the user and the system can initiate interaction, provide information, ask for clarification, signal lack of understanding, or interrupt the other participant. The interface will operate seamlessly across multiple domains -- for example, providing airline schedule information interleaved with weather forecasts for departure and arrival cities -- and will support multiple modalities, including graphics, pointing, and gesture. Underlying the Communicator program is the Communicator architecture, designed to support rapid and cost-effective development

#	Technology	Reference	Description
			of multi-modal human computer interfaces. This ambitious design specifies that the interface components and systems produced by the developers will be required to follow a set of standards that promote interoperability and plug-and-play of similar components. Some of these Communicator-compliant standards will be drawn from the commercial domain, while others will be established by the Communicator program itself, with an eye toward influencing future commercial standards. The initial architecture provides a plug-and-play testbed for the development of HCI components for speech recognition, speech synthesis, dialogue management, language understanding, context tracking, and language generation. As the Communicator program matures, we anticipate other components, such as those for handling graphics and gesture input and output modalities. MITRE has established a Web site (http://fofoca.mitre.org) to allow developers to quickly assemble and test new, architecture-compliant interfaces. The Web-based software repository allows Communicator participants to contribute and to access architecture-compliant modules. A Web-accessible testbed allows developers to plug and play the various components in the repository and thereby coordinate their new component with the other components to provide new interface capabilities. The testbed also provides a demonstration and data-gathering facility for the Communicator project. The DARPA Communicator program draws upon development teams from MITRE and other organizations including Bolt Beranek and Newman (BBN), Carnegie Mellon University, IBM, Microsoft, MIT, and Stanford Research Institute (SRI). The MITRE Communicator team maintains the Communicator testbed and the Communicator architecture specification documentation, processes the Communicator architecture bug reports, and fixes bugs. MITRE assists DARPA in evaluating the interface components and systems developed by team members for compliance with the specified architecture. MITRE is also engaged in development of Communicator-compliant interface applications. At

#	Technology	Reference	Description
			present the MITRE team is developing two advanced systems, the first a gesture and spoken language interface for the Army map-based Maneuver Control System and the second a phone-based interface to support airline travel information for the DARPA Communicator community. Putting different interface components into a robust interface is a major challenge, not only for experienced researchers, but also for application developers who have no intimate familiarity with these components. Some components can be obtained commercially, such as large vocabulary speech recognizers; others, such as dialogue managers, only exist in the laboratory. These components must be adapted for the intended application and assembled into a coherent system. An additional Communicator program goal for the MITRE team is to provide training that will allow developers to assemble advanced human computer interfaces in a reasonable amount of time.
6	Multipe HSI Techrologies	***DARPA Augmented Cognition Initiative*** A new well-funded initiative that promotes development of multimodal interfaces to balance load on sensory channels. Presentation from LCDR Dylan Schmorrow, DARPA/IPTO, July 2001	**Overview** The goal of the *DARPA Augmented Cognition* effort is to extend, by an order of magnitude or more, the information management capacity of the human-computer warfighting integral by developing and demonstrating quantifiable enhancements to human cognitive ability in diverse, stressful, operational environments. Specifically, this effort will empower one human's ability to successfully accomplish the functions currently carried out by three or more

#	Technology	Reference	Description
			individuals. A key objective of the effort is to foster development of novel- and improvement of identifiable- prototypes and enabling technologies, in order to experiment with and understand the means by which they may be integrated into existing operational systems, as well as those in development. The effort will accomplish this by delivering new design principles for human-computer symbiosis. The Augmented Cognition effort will explore the interaction of cognitive, perceptual, neurological, and digital domains to develop improved performance application concepts. The advanced applications will be tailored to military problems in order to demonstrate potential pay-off for operational users. Success will improve the way 21st Century warriors interact with computer based systems, advance systems design methodologies, and fundamentally revolutionize military decision making. Perhaps most important among the enabling breakthroughs are (1) the abundance of laboratory results which are accruing from the cognitive revolution (the science of human problem solving and reasoning) and, (2) the results of research on brain mechanisms based on functional Magnetic Resonance Imaging (fMRI) technology that accrued from the "decade of the brain," funded in large part by the National Institutes of Health. This revolution in human information capacity is further enabled by continued gains in speed and memory growth in digital technologies; success in the miniaturization, powering, and

#	Technology	Reference	Description
			ruggedization of hardware; significant breakthroughs in software-based data mining and storage schemas; and robust input-output methods that blend virtual representations across perceptual modalities. These breakthroughs are currently at the pre- or prototype stage of maturity, they are largely not integrated, and their successes are based principally on laboratory rather than field-testing. Initially the effort will develop the cognitive tools and technological capability to support experiments using the "InfoCockpit" generic testbed concept. The first 18 months of the effort will focus on establishing and demonstrating a performance baseline seeking the range of improvement achievable through immersion, sensory inputs and state. The effort will use current and future military requirements within operational environments to conduct experiments to demonstrate and validate the technologies developed. The final 24 months of the effort will focus on developing and demonstrating prototype technologies with military utility.
7	Speech Interface	***Spacecraft Speech Command Scripting Prototype*** Robert Remington & Howard Coven, "Using a Spoken Language User Interface for Satellite Command Scripting." *Audio Input & Output*	**Abstract** Speech and natural language interface technologies show promise for significantly improving the usability of many interactive applications, including the monitoring and control of valuable commercial and military space-based assets. Based upon an earlier cognitive task analysis performed by one of the authors (Remington), a specific task routinely performed by experienced spacecraft test engineers and operators was often described as being relatively tedious and error-prone. The problematic task involves the creation of scripts, or macros, consisting of a series of commands to test, monitor, or control a spacecraft. The macro scripting task typically involves the use of a specialized commanding language with very cryptic commands and a syntax that is

#	Technology	Reference	Description
		Society 2002 Conference Proceedings, May, 2002	not English-like. As a result, creation of satellite commanding scripts can be a very time-consuming and error-prone task with a relatively long learning curve. Therefore, the command scripting task was singled out as an example of a spacecraft controller task that might benefit from the application of current speech recognition technology to provide a. natural language speech dialog. This paper describes the operational features of a proof-of-concept prototype that provides a compelling demonstration of how the current time-consuming error-prone manual keying of cryptic spacecraft commands might be performed with natural spoken English commands using commercially available low-cost speech recognition technology. The Spacecraft Speech Command Scripting prototype incorporates Dragon Systems *Naturally Speaking*™ product to perform the fundamental speech recognition and text-to-speech synthesis processes. In addition, the ActiveX components provided by *Naturally Speaking*™ SDK are incorporated in our custom Visual Basic code to implement many of the speech interface features. Demonstrations of the prototype can be given on desktop and portable IBM-compatible PCs capable of running Microsoft Windows 2000™. This proof-of-concept prototype replaces the manual keying of cryptic spacecraft commands with natural spoken English commands. For example, the operator can create a script that will control a large number of spacecraft components with plain English speech commands such as "open value one," "rotate platform 40 degrees right," tilt platform 36 degrees forward," and "reset the cooling system temperature." The operator can quickly view all the currently valid interface control commands and the script creation commands simply by saying "What can I say." Voice annotations can be attached to a script by dictating in a normal fashion by taking advantage of the system's continuous speech recognition capability. The system augments normal visual feedback with real-time audio feedback and guidance using speech synthesis. In addition to describing the implementation details associated with each of

#	Technology	Reference	Description
			the speech-enabled features, the present paper provides a discussion of speech user interface design issues and the future plans to develop a more comprehensive implementation of spacecraft control tasks.
8	Multimodal Interface Architecture	***DARPA Communicator Architecture (DCA)*** Alan Goldschen, "The Role of the DARPA Communicator Architecture as a Human Computer Interface for Distributed Simulations."	**Abstract** We describe an emerging architecture, the DARPA Communicator Architecture (DCA), which defines human computer interface (HCI) standards for advanced spoken dialogue systems [1]. We describe how interoperable components specified by the DCA support new HCI capabilities in distributed wargame simulations, such as Modular Semi-Automated Forces (SAF) simulations. The DCA is expected to emerge as an HCI standard for advanced spoken dialogue systems specifically promoting the use of interoperable plug-and-play components. The DCA is expected to define interoperability standards for different spoken dialogue components such as speech recognition, natural language processing, dialogue (discourse) management, natural language generation, and speech synthesis. We anticipate applications (such as distributed simulations) that use the DCA for its HCI to obtain benefits not previously available, such as reducing the time to integrate HCI spoken dialogue components into applications. The application has the opportunity to leverage off HCI component advancements in a variety of emerging research and commercial products with minimal integration efforts using the plug-and-play paradigm. Additionally, the

#	Technology	Reference	Description
			DCA's component framework allows for more flexible higher-level dialogue strategies than is possible with traditional "forward pipeline" architectures. For instance, the discovery of a previously unknown entity in a SAF is disseminated via the DCA for the speech recognition component to dynamically update its vocabulary for later recognition. We describe our work using the DCA to .support a spoken language HCI for an Army-based SAF simulation. We describe the integration effort involved to make this SAF simulation compatible with the DCA. Additionally, we describe how the DCA differs from other SAF system architectures that use HCI components, such as CommandTalk that uses the Open Agent Architecture.
9	Multirrodal Speech/Touch	Bolt, R. A., " Put-that-there: Voice and gesture at the graphics interface." *ACM Computer Graphics*, 14, 3 (1980), 262-270. The "classic" first multimodal research project	**Abstract** Our ability to develop robust multimodal systems will depend on knowledge of the natural integration patterns that typify people's combined use of different input modes. To provide a foundation for theory and design, the present research analyzed multimodal interaction while people spoke and wrote to a simulated dynamic map system. Task analysis revealed that multimodal interaction occurred most frequently during spatial location commands, and with intermediate frequency during selection commands. In addition, microanalysis of input signals identified sequential, simultaneous, point-and-speak, and compound integration patterns, as well as data on the temporal precedence of modes and on inter-modal lags. In synchronizing input streams, the temporal precedence of writing over speech was a major theme, with pen input conveying location information first in a sentence. Linguistic analysis also revealed that the spoken and written modes consistently supplied complementary semantic information, rather than redundant.

#	Technology	Reference	Description
			One long-term goal of this research is the development of predictive models of natural modality integration to guide the design of emerging multimodal architectures.
10	Multimodal Speech & Gesture	Emilio Schapira and Rajeev Sharma, "Experimental Evaluation of Vision and Speech based Multimodal Interfaces." 2002	**Abstract** Progress in computer vision and speech recognition technologies has recently enabled multimodal interfaces that use speech and gestures. These technologies offer promising alternatives to existing interfaces because they emulate the natural way in which humans communicate. However, no systematic work has been reported that formally evaluates the new speech/gesture interfaces. This paper is concerned with formal experimental evaluation of new human-computer interactions enabled by speech and hand gestures. The paper describes an experiment conducted with 23 subjects that evaluates selection strategies for interaction with large screen displays. The multimodal interface designed for this experiment does not require the user to be in physical contact with any device. Video cameras and long range microphones are used as input for the system. Three selection strategies are evaluated and results for different target sizes and positions are reported in terms of accuracy, selection times and user preference. Design implications for vision/speech based interfaces are inferred from these results This study also raises new question and topics for future research. Some formal studies from the human-computer interaction perspective have included experiments

#	Technology	Reference	Description
			to evaluate tasks on multimodal interfaces. The main purpose of this study is to conduct formal experiments to evaluate selection strategies using vision and speech based multimodal interfaces. Figure 1 shows a picture of the "device-free" interface. Due to space limitation, some details have been omitted in this paper; for a complete description of this study, refer to: E. Schapira. "Experimental evaluation of vision and speech based multimodal interfaces." Master's thesis, The Pennsylvania State University, Aug. 2001.
11	Multimodal Speech/Pen	***QuickSet*** Cohen, P. R., et al. "Quickset: Multimodal interaction for distributed applications. "*Proceedings of the Fifth ACM International Multimedia Conference*, ACM Press: 1997 New York, 31-40.	**Abstract** This paper reports on a case study comparison of a direct-manipulation-based graphical user interface (GUI) with the QuickSet pen/voice multimodal interface for supporting the task of military force "laydown." In this task, a user places military units and "control measures," such as various types of lines, obstacles, objectives, etc., on a map. A military expert designed his own scenario and entered it via both interfaces. Usage of QuickSet led to a speed improvement of 3.2 to 8.7-fold, depending on the kind of object being created. These results suggest that there may be substantial efficiency advantages to using multimodal interaction over GUIs for map-based tasks.
12	Multimodal & Tangible Interfaces	***RASA Tangible Interface*** McGee, D. R., Cohen, P. R., Wesson, R. M., & Horman, S.: "Comparing paper and tangible multimodal tools," in the Proceedings of the Conference on Human Factors in Computing	**Abstract** In command posts, officers maintain situational awareness using paper maps, Post-it notes, and hand-written annotations. They do so because paper is robust to failure, it is portable, it offers a flexible means of capturing information, it has ultra-high resolution, and it readily

#	Technology	Reference	Description
		Systems (CHI'02), ACM Press, Minneapolis, April, 2002.	supports face-to-face collaboration. We report herein on an evaluation comparing maps and Post-its with a tangible multimodal system called ***Rasa***. Rasa augments these paper tools with sensors, enabling it to recognize the multimodal language (both written and spoken) that naturally occurs on them. In this study, we found that not only do users prefer Rasa to paper alone, they find it as easy or easier to use than paper tools. Moreover, Rasa introduces no discernible overhead in its operation other than error repair, yet grants the benefits inherent in digital systems. Finally, subjects confirmed that by combining physical and computational tools, Rasa is resistant to computational failure.
13	Speech Interface	***Integrated Network Management System (INMS)*** Robert Remington, "Spoken Language Interface for a Network Management System." *Proceedings of the International Society for Optical Engineering (SPIE), Multimedia Systems & Applications,* September 1999, Boston, MA	**Abstract** Leaders within the Information Technology (IT) industry are expressing a general concern that the products used to deliver and manage today's communications network capabilities require far too much effort to learn and to use, even by highly skilled and increasingly scarce support personnel. The usability of network management systems must be significantly improved if they are to deliver the performance and quality of service needed to meet the ever-increasing demand for new Internet-based information and services. Fortunately, recent advances in spoken language (SL) interface technologies show promise for significantly improving the usability of most interactive IT applications, including network management systems. The emerging SL interfaces will allow users to communicate with IT applications through words and phases – our most familiar form of everyday communication. Recent advancements in SL technologies have resulted in new commercial products that are being operationally deployed at an increasing rate.

#	Technology	Reference	Description
			The present paper describes a project aimed at the application of new SL interface technology for improving the usability of an advanced network management system. It describes several SL interface features that are being incorporated within the *Integrated Network Management System*, an existing system with a modern graphical user interface (GUI), including 3-D visualization of network topology and network performance data. The rationale for using these SL interface features to augment existing user interfaces is presented, along with selected task scenarios to provide insight into how a SL interface will simplify the operator's task and enhance overall system usability. Many commercial and military automated monitoring and supervisory control applications (e.g., satellite constellation management, air traffic control, and military command and control centers) have user interface requirements that are very similar to those of the network management application described in this paper. It is anticipated that the application and .validation of emerging SL interface technologies along the lines reported here will lead to a significant improvement in the usability of future network management and similar systems.
14	Automatic Speech Recognition	Jan M. Noyes, et. al., "Automatic Speech Recognition, Noise And Workload." *Proceedings of the Human Factors and Ergonomic Society*, 2000.	**Abstract** Despite the increasing use of technology in the developed world, most computer communications still take place via a QWERTY keyboard and a mouse. The use of Automatic Speech Recognition (ASR) whereby individuals can 'talk' to their computers has yet to be realized to any great extent. This is despite the benefits relating to greater efficiency, use in adverse environments and in the 'hands-eyes busy' situation. There are now affordable ASR products in the marketplace, and many people are able to buy these products and try ASR for themselves. However, anecdotal reports suggest that these same people will use ASR for a few days or weeks and then revert to conventional interaction techniques; only a hardy few appear to persist long enough to reap the benefits. Thus, it is our contention that ASR is a commercially viable

#	Technology	Reference	Description
			technology but that it still requires further development to make a significant contribution to usability. Admittedly, there are some very successful applications that have used ASR for a number of decades, but these are often characterized by relatively small vocabularies, dedicated users and non-threatening situations; typical applications are in offices (Noyes & Frankish, 1989) or for disabled users (Noyes & Frankish, 1992). Given that Armoured Fighting Vehicles (APVs) could employ ASR with limited vocabulary and dedicated users, the use of ASR in this application is considered here. The principle difference between ASR for APV and previous applications is the environmental conditions in which the technology will be used.
15	Multimodal Speech/Gesture	Sharon Oviatt, et al. "Designing the User Interface for Multimodal Speech and Pen-based Gesture Applications: State-of-the-Art Systems and Future Research Directions" *Human-Computer Interaction* August, 2000	**Abstract** The growing interest in multimodal interface design is inspired in large part by the goals of supporting more transparent, flexible, efficient, and powerfully expressive means of human computer interaction than in the past. Multimodal interfaces are expected to support a wider range of diverse applications, to be usable by a broader spectrum of the average population, and to function more reliably under realistic and challenging usage conditions. In this paper, we summarize the emerging architectural approaches for interpreting speech and pen-based gestural input in a robust manner— including early and late fusion approaches, and the new hybrid symbolic/statistical approach. We also describe a diverse collection of state-of-the-art multimodal systems that process users' spoken and gestural input. These applications range from map-based and virtual reality systems for engaging in simulations and

#	Technology	Reference	Description
			training, to field medic systems for mobile use in noisy environments, to web-based transactions and standard text-editing applications that will reshape daily computing and have a significant commercial impact. To realize successful multimodal systems of the future, many key research challenges remain to be addressed. Among these challenges are the development of cognitive theories to guide multimodal system design, and the development of effective natural language processing, dialogue processing, and error handling techniques. In addition, new multimodal systems will be needed that can function more robustly and adaptively, and with support for collaborative multi-person use. Before this new class of systems can proliferate, toolkits also will be needed to promote software development for both simulated and functioning systems.
16	Multimodal Speech/Pen	Sharon Oviatt. "Multimodal Signal Processing in Naturalistic Noisy Environments." *Proceedings of the International Conference on Spoken Language Processing*, August, 2000, Beijing China	**Abstract** When a system must process spoken language in natural environments that involve different types and levels of noise, the problem of supporting robust recognition is a very difficult one. In the present studies, over 2,600 multimodal utterances were collected during both mobile and stationary use of a multimodal pen/voice system. The results confirmed that multimodal signal processing supports significantly improved robustness over spoken language processing alone, with the largest improvement during mobile use. The multimodal architecture decreased the spoken language error rate by 19-35%. In addition, data collected on a command-by-command basis while users were mobile emphasized the adverse impact of users' Lombard adaptation on system processing, even when a noise-canceling microphone was used. Implications of these findings are discussed for improving the reliability and stability of

#	Technology	Reference	Description
			spoken language processing in mobile environments.
17	Multimodal Speech/Touch Tablet	Sharon Oviatt. "Preferred modalities in dialogue systems," *Proceedings of the International Conference on Spoken Language Processing*, August, 2000, Beijing, China, Vol. II, 727-730.	**Abstract** This research describes which modalities are preferred in particular contexts when interacting with a multi-modal dialogue system. The trade-off between three factors is investigated: (*i*) speech recognition performance, (*ii*) efficiency of input modality and (*iii*) the system's output modality. Four versions were developed of a multi-modal examinator to be used in elementary school. The versions differed in recognition performance ('perfect' vs. realistic) and output modality (speech or text). In all systems, subjects could provide input via speaking or typing. Answer length in characters was used as a measure of efficiency. Results show that both speech recognition performance and efficiency have a strong impact on preferred modalities. No effect was found of the system's output modality. In particular, he found that *short* input (limited number of characters) is more efficiently entered via keyboard, and arguably this may influence modality selection. This might be related to the general tendency in humans (observed by e.g., Zipf 1949) to opt for minimal effort. The experiments of Oviatt et al. were done with a simulated "Service Transaction System", which can assist users with such tasks as renting a car or personal banking. Subjects had the possibility of to enter information by voice or by writing (on an LCD tablet). It is worth noting that subjects did not really engage in a *dialogue* A strong influence of speech recognition performance on modality selection was found. When being examined by a system with a realistic recognizer, subjects type significantly more often than they do when being questioned by systems with a perfect recognizer. The majority of the modality switches is related to speech recognition errors. This is in line with the findings of Oviatt and co-workers. Moreover, it was found that numerical answers are somewhat more likely to be typed than other answers. We feel that this is related to efficiency, because the [illegible]ical

#	Technology	Reference	Description
			answers could always be entered with a highly limited number of keystrokes. However, on the basis of this experiment we can not rule out the possibility that Oviatt's contrastive functionality was at work. The experiment does provide evidence for the influence of efficiency on modality selection, since subjects generally prefer speech as input modality and indicate that they do so on the basis of speech being faster and more easy to use.
18	Multimodal Interface versus Speech-only interface	Sharon Oviatt, "Mutual Disambiguation of Recognition Errors in a Multimodal Architecture," *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99)*, ACM Press: New York, 576-583.	**Abstract** As a new generation of multimodal/media systems begins to define itself, researchers are attempting to learn how to combine different modes into strategically integrated whole systems. In theory, well designed multimodal systems should be able to integrate complementary modalities in a manner that supports mutual disambiguation (MD) of errors and leads to more robust performance. In this study, over 2,000 multimodal utterances by both native and accented speakers of English were processed by a multimodal system, and then logged and analyzed. The results confirmed that multimodal systems can indeed support significant levels of MD, and also higher levels of MD for the more challenging accented users. As a result, although speech recognition as a stand-alone performed far more poorly for accented speakers, their multimodal recognition rates did not differ from those of native speakers. Implications are discussed for the development of future multimodal architectures that can perform in a more robust and stable manner than individual recognition technologies. Also discussed is the design of interfaces that support diversity in tangible ways, and that function well under challenging real-world usage conditions.
19	Multimodal	***Sensory Interfaces Program***	**Overview** Researchers at the Integrated Multimedia Center (IMSC) are

#	Technology	Reference	Description
	Speech & Haptics	***Program*** Integrated Multimedia Systems Center (IMSC) University of Southern California. Sponsored by the NSF Overview is based upon live demonstrations given at Scientific Advisory Board tours of the IMSC.	investigating multimodal sensory interfaces as they relate to immersive and partial immersive environments. One area of multimodal sensory interface research focuses on haptics. Haptics involves the modality of touch and the sensation of shape and texture an observer feels when exploring a virtual object, such as a three-dimensional model, a tactile map, or a graphic designer's rendering of an imaginary object. The Seismic Information Map is an example of one of the projects that incorporates speech input and haptics. It involves a rendered 3D map of the Los Angeles basin, suitable for haptic exploration with the haptic devices including the PHANToM haptic glove. In this project, they integrate haptics with a seismic map and speech recognition. The earthquake data is rendered haptically is of three ways (1) a depth map of the Los Angeles basin, showing the underlying bedrock that forms the bottom of the basin; (2) a map of the surface geology of the area, which shows where the mountains are and where the soil is likely to liquefy; (3) a map of the shaking in various parts of the area due to specific earthquake events (e.g., Northridge). The most interesting map to render haptically is of the third type. The data are available in the form of amplitude of shaking as a function of time. These are rendered to the tip of the PHANToM haptic glove device. The first kind of data (depth) is rendered by giving the map more deformability in areas of greater depth (so the user can push down further. Using speech recognition a user can interact with the map via spoken commands.
20	Multimodal Interfaces	***Multimodal Interfaces*** Human Interface Technology (HIT) Lab, University of Washington	**Abstract** This project (Multimodal interfaces) involves the development of software libraries for incorporating multimodal input into human computer interfaces. These libraries combine natural language and artificial intelligence techniques to allow human computer interaction with an intuitive mix of voice, gesture, speech, gaze and body motion. Interface designers will be able to use this software

#	Technology	Reference	Description
			for both high and low level understanding of multimodal input and generation of the appropriate response. Intelligent Conversational Avatar – the purpose of this project is to develop an expert system and natural language parsing module to parse emotive expressions from textual input. GloveGRASP – a set of C++ class libraries that allow developers to add gesture recognition to their SGI applications. HRMS – a project to develop a generic software package for hand motion recognition using hidden markov models, with which user interface designers will be able to build a multimodal input system.
21	Spatial Audio	Nelson, T.W., et al. "Spatial Audio Displays For Speech Communications: A Comparison Of Free Field And Virtual Acoustic Environments." *Proceedings of the Human Factors and Ergonomic Society*, 2000	**Abstract** The ability of listeners to detect, identify, and monitor multiple simultaneous speech signals was measured in free field and virtual acoustic environments. Factorial combinations of four variables, including audio condition, spatial condition, the number of speech signals, and the sex of the talker were employed using a within-subjects design. Participants were required to detect the presentation of a critical speech signal among a background of non-signal speech events. Results indicated

#	Technology	Reference	Description
			that spatial separation increased the percentage of correctly identified critical speech signals as the number of competing messages increased. These outcomes are discussed in the context of designing binaural speech displays to enhance speech communication in aviation environments.
22	Spatial Audio	Robert S. Bolia, et al. "A Speech Corpus For Multitalker," *Communications Research Acoustical Society of America. 2000*	**Abstract** Several recent experiments at the Air Force Research Laboratory have investigated the utility of spatial audio displays for augmenting speech intelligibility in multitalker communications environments - Bolia *et al.*, 1999; Nelson *et al.*, 1998a; Nelson *et al.*, 1998b; Simpson *et al.*, 1999!. Some of the goals of this research included: 1) an empirical determination of the maximal number of channels for which the benefits of spatialization may be realized in a spatial audio display designed to aid in the segregation of simultaneous, context-independent speech sources; 2) an evaluation of the efficacy of four different spatialization schemes for this task; and 3) the manner in which these factors interact with the sex of the target talker. In order to accomplish these goals, a large number of speech samples from talkers of both sexes were required. The purpose of this article is to describe the methods employed in collecting these speech samples, as well as the form of the resulting corpus, with the intent that other researchers in the field might benefit from their availability.

#	Technology	Reference	Description
			A database of speech samples from eight different talkers has been collected for use in multitalker communications research. Descriptions of the nature of the corpus, the data collection methodology, and the means for obtaining copies of the database are presented.
23	Spatial Audio	***Virtual Auditory Space*** Researchers in AFRL's *Human Effectiveness Directorate,* in collaboration with Dr. Barbara Shinn-Cunningham of Boston University,	**Abstract** Information overload is an ever-increasing concern for human performance, both in and outside the military environment. In air and ground operations, the sheer amount and complexity of information can exceed the amount a person can comprehend, especially in high-tempo, critical missions. We have made important strides in solving this problem for headphone users. The goal is to create headphone-based auditory displays in which each sound source has a well-defined position in three-dimensional space. The first step is to isolate the various electronic signals reaching a headphone so that each communication occupies one channel in a multi-channel system. Potential applications abound. For example, a pilot could hear the AWACS controller speak from one consistent position in the cockpit, while the navigator always speaks form another. Signals from the ground could appear to come below. Signals from another aircraft could correlate with the craft's actual position. An "auditory pointer" could

#	Technology	Reference	Description
			be adapted to improve directional awareness for hearing-aid users. In each case, the goal is to harness direction-selective listening to increase human information-processing capacity.
24	Eye & Gesture Tracking	Vildan Tanriverdi and Robert J.K. Jacob. Department of Electrical Engineering and Computer Science Tufts University	**Excerpts** Eye movement-based interaction is an example of the emerging non-command based interaction style. In this type of interaction, the computer observes and interprets user actions instead of waiting for explicit commands. Interactions become more natural and easier to use. One system that suggests such advantages is a screen-based system developed by Starker and Bolt. It monitors eye movement-based interaction offers the potential of easy, natural, and fast ways of interacting in virtual environments. However, there is little empirical evidence about the advantages or disadvantages of this approach. We developed a new interaction technique for eye movement interaction in a virtual environment and compared it to more conventional 3-D pointing. We conducted an experiment to compare performance of the two interaction types and to assess their impacts on spatial memory of subjects and to explore subjects' satisfaction with the two types of interactions. We found that the eye movement-based interaction was faster than pointing, especially for distant objects. However, subjects' ability to recall spatial information was weaker in the eye condition than the pointing one. Subjects reported

#	Technology	Reference	Description
			equal satisfaction with both types of interactions, despite the technology limitations of current eye tracking equipment.
25	Eye Tracking & Gesture Recognition	***IBM Blue Eyes Project*** IBM Almaden Research Center in San Jose, CA Demonstration viewed at IBM Make It Easy Conference July, 2001 at IBM Almaden Research Center. Website http://www.almaden.ibm.com/cs/blueeyes/find.html	**Excerpts** IBM's *BlueEyes* uses non-obtrusive sensing technology, such as video cameras and microphones, to identify and observe a user's actions, and to extract key information, such as where the user is looking and what the user is saying verbally and gesturely. These cues are analyzed to determine the user's physical, emotional, or informational state, which in turn can be used to help make the user more productive by performing expected actions or by providing expected information. IBM researchers at their Almaden Research Center discovered a fast, robust, and low cost pupil detection technique that uses two infra red (IR) time multiplexed light sources, composed of two rings of 8 LED's each, synchronized with the camera frame rate. One light source is placed very close to the camera's optical axis, and the second source is placed off-axis. The pupil appears bright in the camera image during on-axis illumination (similar to the red eye effect from flash photography), and dark when illumination is off-axis. The off-axis light source is calibrated to provide roughly equivalent whole-scene illumination. Pupil detection follows from thresholding the difference of the dark from the bright pupil images. To reduce artifacts caused mostly by head

#	Technology	Reference	Description
			motion, a larger temporal support is used. This method can be applied to detect and track several pupils (or several people). Experimental results from a real-time implementation of the system show that this technique is very robust, and able to detect pupils using wide field of view low cost cameras under different illumination conditions, even for people with glasses, and up to 5m from the camera.
25	Eye Tracking	Linda E. Sibert and Robert J. K. Jacob", "Evaluation of eye gaze interaction," *CHI 2000 Proceedings*, pages 281-288.	**Abstract** Eye gaze interaction can provide a convenient and natural addition to user-computer dialogues. We have previously reported on our interaction techniques using eye gaze. While our techniques seemed useful in demonstration, we now investigate their strengths and weaknesses in a controlled setting. In this paper, we present two experiments that compare an interaction technique we developed for object selection based on a where a person is looking with the most commonly used selection method using a mouse. We find that our eye gaze interaction technique is faster than selection with a mouse. The results show that our algorithm, which makes use of knowledge about how the eyes behave, preserves the natural quickness of the eye. Eye gaze interaction is a reasonable addition to computer interaction and is convenient in situations where it is important to use the hands for other tasks. It is particularly beneficial for the larger screen workspaces and

#	Technology	Reference	Description
			virtual environments of the future, and will be increasingly practical as eye-tracker technology become more mature.
27	Gesture Recognition	***GloveGRASP*** Human Interface Technology (HIT) Lab, University of Washington http://www.hitl.washington.edu/research/multimodal/GestureGRASP.html	**Overview** GloveGRASP is a C/C++ class library that allows software developers to add highly accurate gesture recognition to their SGI code. Glove GRASP uses advanced pattern matching techniques to ensure highly accurate gesture recognition. Users can set the parameters used in the pattern matching to tune the recognition for their particular gesture set. Recognition rates of more than 95% can be achieved with gesture sets containing as many as a dozen separate gestures. Unlike other gesture recognition systems, GloveGRASP uses context dependent feature based recognition. This means that the same gesture may be interpreted differently for various interaction contexts – so a small gesture set can provide a large range of commands across all the possible interaction contexts. Contexts and the symbols representing each gesture are completely user definable. Allowing context dependent recognition reduces the need to remember dozens of different gestures and produces a very high recognition rate. The GloveGRASP package includes the following features: 1. 5DT 5^{th} Glove SGI device drivers.

#	Technology	Reference	Description
			2. User dependent gesture training. 3. One or two handed gesture recognition. 4. Real-time feature-based continuous & discrete gesture recognition. 5. Context dependent recognition & dynamic switching of training sets. 6. Inter-process communications through UNIX TCP-IP Sockets.
28	3D Visualization	***Satellite Operations Simulator (SOpSim)*** AFR- Williams, Mesa research project headed by Capt. Chris Biegun Overview based upon discussions with Capt. Biegun and live demonstration	**Overview** The objective of this project is to develop a satellite operations training and rehearsal research test bed. The primary focus at the outset was on developing a Space Maneuver Vehicle (SMV) simulator for use in orbital mechanics training. The SOpSim control console interface design was influenced by lessons learned from the Mesa Predator UAV trainer. One of the control consoles key HSI features is the use of 3D presentation techniques to provide realistic high-fidelity visualization of satellite orientation in space from basic maneuvering to on-orbit servicing maneuvers. Future visualization enhancements include high-fidelity visual of sun, moon, stars, penumbra, and vehicle lights. Planned human-centered research will be used to guide space system design and development concerning: • Data and interface needs for situation awareness and workload management in space

#	Technology	Reference	Description
			• Human performance models of complex behavior • Data-driven analysis of alternative training/rehearsal methods • Identify and address operator problems prior to system delivery • Assessments of air model for space operations While current plans do not call for the incorporating alternative control technologies such as speech and gesture, Capt. Biegun expressed an interest in such HSI technologies with their potential to improve maneuvering in 3D space. The problems associated with maneuvering objects in 3D space with the traditional 2D mouse and trackball devices are well-known to HSI researchers.
29	Virtual Retinal Displays for HMDs	Human Interface Technology (HIT) Lab, University of Washington	**Overview** The Virtual Retinal Display (VRD) team has been focused on developing improvements to the current prototype systems and on creating the parts needed for future prototypes. The VRD, based on the concept of scanning an image directly on the retina of the viewer's eye, was invented at the HIT lab in 1991. The development program began in November 1993, with the goal of producing a full color, wide field-of-view, high resolution, high brightness, low cost virtual display. Participating in the integrated Small Precision Optics

#	Technology	Reference	Description
			Manufacturing Technology (ISPOMT) consortium, the HIT lab is designing and developing an interactive VRD for US Navy pilots. The VRD is the only display technology that has sufficient luminance to be used as an augmented display over the pilot's real worldview in bright sunlight. The HIT lab will develop this unique augmented image as an interactive display. New technologies that will be developed are: 1. Exact registration of the augmented image over the real world scene. 2. Eye tracking to know where the pilot is looking in the augmented image, and Display strategies that both allow .the pilot to view sensor data from 360 degrees and to prompt the display for additional information.
30	Helmet Mounted Displays (HMDs)	"Head-Slaved Tracking In A See-Through Hmd: The Effects Of A Secondary Visual Monitoring Task On Performance And Workload," W. Todd Nelson, Robert S. Bolia, Chris A. Russell Air Force Research Laboratory Wright-Patterson AFB, Ohio Rebecca M. Morley and Merry M. Roe Sytronics Inc. Dayton, Ohio, *Human Factors And Ergonomics Society 44th Annual Meeting*, 2000.	**Abstract** Technological advances in helmet-mounted displays (HMDs) have permitted the design of "see-through" displays in which virtual imagery may be superimposed upon real visual environments. The utility of see-through displays in multitask environments remains uncertain, especially in environments that involve switching one's attention between those tasks represented in the virtual display and those existing in the real world. The present study was designed to assess the effects of a secondary visual monitoring task on performance and workload in a head-slaved tracking task. Participants attempted to center a reticle over a moving circular target using a Kaiser Electronics SimEye 2500 HMD while concurrently performing the visual monitoring task component of the Multi-Attribute Task Battery (MATB; Comstock & Arnegard, 1992), which was displayed on a computer monitor. Task difficulty for the head-

#	Technology	Reference	Description
			slaved tracking task was varied by manipulating time delay. Results are discussed in terms of their implications for practical implementation of see-through HMDs in multi-task environments.
31	Multiple HSI Technologies	***Virtual Cockpit Optimization Program*** Program Office Web-site. http://134.78.40.107/VCOP/page1.htm	**Abstract** The Virtual Cockpit Optimization Program (VCOP) is providing an answer to the problem of information overload for pilots of modern military aircraft while reducing the cost of upgrading legacy aircraft. *The concept of the virtual cockpit program is to provide the pilot with information such as situational awareness, sensor imagery, flight data, and battlefield information in a clear, non-confusing and intuitive manner, thus making the aircraft easier and safer to fly while also improving mission performance.* The majority of the VCOP activity involves the integration of advanced technologies into a single system that represents a significant leap ahead in cockpit design philosophies. Rather than concentrating on the aircraft and how it can be retrofitted to meet the needs of the next generation warfighter, VCOP furnishes pilots with the necessary enhanced capabilities to perform their job more efficiently. VCOP is comprised of the following five independently developed technologies: 1. Full color, high resolution, high brightness helmet-mounted display (HMD) that incorporates Virtual Retinal Display (VRD) technology 2. Three Dimensional (3D) audio 3. Speech recognition

#	Technology	Reference	Description
			4. Intelligent information management 5. Crew-aided cognitive decision aides These technologies are being integrated into a single approach based on evolutionary principles developed through Department of Defense (DoD) and industry initiatives called Simulation Based Acquisition (SBA). The concept behind SBA is to integrate Modeling and Simulation (M&S) tools and technologies into the acquisition process to provide a higher quality product at a lower cost and in a shorter amount of time than traditional methods. The goals of SBA are to: 1. Substantially reduce the time, resources, and risk associated with the acquisition process; 2. Increase the quality of the resulting product while reducing Total Ownership Costs (TOC) throughout the system life cycle; and 3. Enable Integrated Product and Process Development (IPPD) throughout the life cycle of the acquisition process [1]. SBA for the Army however, applies to more than just the acquisition community. To provide the soldier with high quality systems in a cost and time efficient manner, M&S tools must be extended to the requirements and training communities as well. In an initiative called Simulation and Modeling for Acquisition, Requirements, and Training (SMART), the Army extends the SBA principles to the Advanced Concepts Requirements (ACR) and Training, Exercises & Military Operations (TEMO) domains (see figure

#	Technology	Reference	Description
			1) [2]. By utilizing the SMART concept with existing technologies, VCOP is furnishing the Vertical Take-Off and Landing (VTOL) community with a highly sophisticated pilot-platform interface that will enhance pilot performance and safety while minimizing the schedule, cost, and technical risk inherent in upgrading legacy aircraft.
32	Virtual Reality HSI Technologies	***Virtual Command Center*** Dockery and Hill. *'Virtual Command Center'*, presented at the *Third International Symposium on C2 Research and Technology*, National Defense University, Washington, D.C., June 1997.	**Excerpts** It is our objective to demonstrate with this project that Virtual Reality (VR) technologies are the next developmental step in command and control (C2) systems. While weapons systems throughout the U.S. armed forces have been modernized for use in Cyberspace, the current C2 infrastructure lags behind. The Virtual Command Center (VCC) software prototype features threshold VR capabilities geared toward modernizing C2 without the large investment in terms of dollars, equipment, and training usually associated with VR applications. The VCC, as described in this paper, gives a glimpse of how the C2 infrastructure can operate in Cyberspace. The past has seen command staffs characterized by large numbers of personnel, centrally located, and operating in a hierarchical construct. Though recent advances in computers and communications have automated many of the individual functions, command staffs are still dependent on centralized manual labor to coordinate staff elements.

#	Technology	Reference	Description
			Changing this paradigm is the key to modernizing the command staff of the future. Virtual Reality transforms human sensory operations from the real world to virtual; computer generated worlds such that the user has a "sense of being there." [Durlach, 1997] Virtual Reality will allow distributed commanders and their staffs to behave and operate as if they were in the same room. Commanders and their staffs can enter VR worlds and perform tasks, collaborate with other participants, and interact with any form of digitized data. Additionally, VR can add visualization techniques to data elements allowing commanders and their staffs to "look and understand" information in a more rapid and intuitive fashion. Virtual Reality development packages were limited, expensive and involved steep learning curves to master their programming techniques. Today, VR is readily available as commercial-off-the-shelf (COTS) development packages that run on common PCs and operating systems. Virtual Reality worlds can be found on the Internet and are accessible by the ordinary user's desktop. Virtual Reality technologies are for today and are viable and affordable solutions for the next generation of U.S. military C2 infrastructure. Implementation: The VCC was developed using WorldToolKit® and World2World® software from Sense8 Corporation. WorldToolKit is a software development package utilizing function libraries and C/C++ code. It supports networked distributed simulations (multicast or IP

#	Technology	Reference	Description
			based) and interfaces devices such as head-mounted-displays (HMDs), trackers, and navigation controllers. World2World^{TM} software provided the client/server environment so that we could run the application on the Internet. The "look and feel" of a remote command center is captured by the surrounding outdoor scene and the field command tent. A tactical sandtable, doors and three viewing screens capture traditional command center functions. The sandtable is a 3-D interactive viewing space for battlefield data. The doors lead to other VR worlds; web sites or displays various form of data. The viewing screens in the rear of the tent show live CNN, the Global Command and Control System (GCCS), and a notional message board. Two human looking sentries are located on each side of the entrance to the tent. These are called avatars. Avatars represent VCC functions or, more importantly, remote users that have entered the VCC to collaborate with other participants. In addition to the field command tent, participants have access via the sandtable to another VR world depicting a virtual. The "look and feel" of a remote command center is captured by the surrounding outdoor scene and the field command tent. A tactical sandtable, doors and three viewing screens capture traditional command center functions. The sandtable is a 3-D interactive viewing space for battlefield data. The doors lead to other VR worlds; web sites or displays various forms of data. The viewing screens in the rear of the tent show live CNN, the Global Command and Control System (GCCS),

#	Technology	Reference	Description
			and a notional message board. Two human looking sentries are located on each side of the entrance to the tent. These are called avatars. Avatars represent VCC functions or, more importantly, remote users that have entered the VCC to collaborate with other participants. In addition to the field command tent, participants have access via the sandtable to another VR world depicting a virtual battlefield where they can visualize military maneuvers and make planning decisions. Data Visualization and Interaction: Within the VCC tents are areas where data are presented. Using the mouse to activate the weather door downloads current weather maps from a weather homepage off the World Wide Web (WWW). The maps appear as standard image files overlaying the weather door. It is our intent to change this feature to allow direct access to user specified weather homepages on the WWW. Additional viewing screens are located in the rear of the VCC tent. Live broadcast news feed from CNN is projected on a virtual television screen. The Global Command and Control System (GCCS) Common Operational Picture (COP) is an NT-based application that is launched and displayed from within the VCC. Loading data, viewing functions such as zoom and highlighting areas of the GCCS COP are available to participants via the mouse. Figure 3 shows a screen capture of these views Virtual Reality is more than a visualization technique. Behind

#	Technology	Reference	Description
			the picture is a computer network that brings databases of information and CPU cycles to the VR environment. Thus, the VR environment can take on functionality normally performed by staff. We have investigated the idea of offloading functionality onto the VCC and found that integrating intelligent agents is viable. For instance, electronic mail functions could be performed by intelligent agent software observing a commander's mail activity and present mail functions via voice communications in the form of a functional avatar. Another example of integration would be to have an intelligent agent monitoring CNN, looking for and then notifying commanders of important information. The VCC seeks to relieve the information overload military commanders experience today by taking full advantage of the intuitive processing of information through visualization techniques. Tactical Sandtable: The VCC has moved the visualization of maps from two-dimensional to multi-dimensional environments that participants can feel a part of, move around in, and manipulate. The tactical sandtable located in the center of the VCC tent displays a two-dimensional map with model tanks that participants can reposition by clicking and dragging the mouse. The sandtable also employs hotspots that change the participant's location such that they are in a VR world of the map features and data elements. Figure 4 shows a view of the sandtable. River World: From a hotspot on the tactical sandtable

#	Technology	Reference	Description
			participants are placed in "River World" which is a VR simulation of the battlefield. Figures 5, 6 and 7 show several views of the "River World". A sample scenario is simulated where friendly forces are attempting to cross a river under various adverse conditions. Participants can "fly" through the terrain examining its features from different perspectives, manipulate the river's water level, and choose from a menu of crossing options for the tanks including fording, bridging, and ferrying. Additionally, uncertainty ab[illegible]e accuracy of the data elements is depicted. The use of colored, translucent domes shade the tanks showing uncertainty about their number or position. Different colors and thickness of fog over the entire battlespace shows uncertainty about weather conditions, terrain features, or how long since the last update of data elements. The "River World" affords opportunities to interact with physical and spatial data by allowing participants to move around inside data or information rather than just viewing it remotely. VR offers users an enhanced interaction with computer generated simulations of the real world. This interaction is both subjective and spatial. Users become immersed in VR worlds where they can take advantage of their natural cognitive and perceptual abilities. Spatial immersion requires the user to get inside 3D space where objects appear to exist and activities occur all around, above, below and in all directions. Special equipment is required to achieve this immersive "feel" in the form of head-mounted display (HMD) systems or to a lesser degree stereoscopic

#	Technology	Reference	Description
			3D glasses. The VCC takes advantage of these immersive technologies. In the "River World," users wearing HMDs gain a sense of presence at the river. The HMD permits a greater degree of depth perception such that users feel they can touch objects like the bridge or a tank. Of course, the use of a virtual reality glove can make this possible. Though not implemented in "River World", virtual reality gloves can allow users to move objects and even initiate functions with the snap of a finger. Immersion greatly enhances the interaction between users and their systems.

Appendix 8 – Evaluation of Advanced Human-System Interface

Appendix 8 – Evaluation of Advanced Human-System Interface

Data Collection Using the Satellite Operator Testbed

Contract No. F33615-00-C-6006

16 July 2002

Prepared by

Thomas J. Sharkey and Robert J. Remington
Monterey Technologies, Inc.
24600 Silver Cloud Court, Suite 103
Monterey, CA 93940-6536

Prepared for
Air Force Research Laboratory
ATTN: John Ianni
2255 H. Street, Building 248
Wright-Patterson AFB, OH 45433-7022

Monterey Technologies, Inc.

ABSTRACT

In order to demonstrate the test bed's capabilities as a human-system interface (HSI) research tool, an experienced satellite controller performed seven support scenarios. Each scenario was performed in two conditions: "baseline" and "enhanced." The HSI features available to the controller differed between these conditions. Objective measures of overall effectiveness were computed from the data collected during these simulations. Following completion of all scenarios, subjective ratings and comments were obtained from the controller. The results and ratings suggest that the enhanced HSI was a marked improvement over the baseline HSI. More importantly, this effort demonstrates that the test bed is fully capable of supporting HSI research in the satellite control setting.

Table Of Contents

Data Collection Using the Satellite Operator Testbed

Introduction

The ability of satellite controllers to quickly and accurately perform a satellite support is, in part, a function of the quality of the human-system interface (HSI) of that workstation. Many operational workstations are designed so that multiple operators are required. In these workstations, the personnel check and double-check each other's entries. The purpose of all this checking is to insure that the satellite is not damaged by erroneous commands. This approach has been with us since the beginning of the space age. The problems with this type of HSI include:

- High number of staff required
- High initial training cost
- High recurrent training cost

It is unlikely that the Air Force, who fly most of the satellites, can continue to invest in the personnel costs that are required to support this operational approach. Scare resources would be better spent in other ways, many believe. Therefore, means to reduce the costs of supporting satellites need to be explored.

The Center for Research Support (CERES) at the Air Force's National Research Center at Schriever AFB, CO and the Air Force Research Laboratory (AFRL) at Wright-Patterson AFB are two organizations that have identified the HSI as being an impediment to reducing the training and other personnel costs. CERES has played a large role in developing a controller workstation that allows a single controller to perform a satellite support. The HSI at CERES is the COTS Based Real Time Architecture (COBRA). This HSI features a windows, mouse, pointer interface similar to those used on commercial workstations. The controllers execute pass plans that contain only validated commands. These pass plans are selected by the satellite engineers based on their analysis of telemetry. In some cases, the controllers select and execute pass plans based on the state of the satellite. For example, if a pressure is nearing a lower limit, the controller has the latitude to execute the pass plan for increasing the pressure without obtaining authorization.

The COBRA HSI and the operational philosophy in place at CERES allow a single controller to perform the vast majority of supports. Rarely another person is required. On those occasions, the second person at the console is often not a controller, but instead a satellite engineer or orbit analyst who provides special expertise.

AFRL has also identified the HSI used to control a satellite as being an area where improvements are warranted. AFRL's interests include the development of HSIs that support tasks not yet being performed with operational satellites. On-orbit servicing is one such task. They also are interested in the ability of a HSI to support tasks that require increased tasking flexibility for satellites. This concept ranges from changing the target area being viewed by the satellite's

sensors, to being able to restore operational capability of a satellite quickly after the satellite has been attacked. In the later case, the satellite controller's detection of anomalies may be the first indication of an attack on that satellite.

In this program, Monterey Technologies, Inc. (MTI) was tasked to develop a test bed in which new HSI concepts could be evaluated. This work, which was funded as part of the Small Business Innovation Research (SBIR) Program, lead to the development of a multi-modal HSI and a test bed in which this HSI and other HSI concepts could be evaluated. The HSI developed by MTI uses the COBRA HSI as a baseline. This report describes a data collection exercise in which a single controller performed a series of scenarios using two levels of the multi-modal HSI. These levels differed in terms of the features available for use by the controller. The primary objective of this effort was to demonstrate the suitability of the test bed as a research setting. A secondary objective was to gather objective information on the effectiveness of MTI's HSI concept.

Method

<u>CONDITIONS</u>

Two HSI conditions were evaluated in this study. We will call these the baseline and enhanced conditions.

<u>Baseline</u>. The baseline condition was modeled after the COBRA interface in use at CERES. This interface requires the controller to scan pages of alphanumeric data in order to identify any variables that are outside their nominal range. Control actions are performed using pass plans.

The baseline system contained a number of HSI features not available in the COBRA system. Key differences include:

1. **Continuous Monitoring of Measurands**. If all of the measurands are within their normal limits, a green circle was displayed on the control panel. If one or more variables were outside of the normal range, then the green circle was removed and a yellow square and/or a red triangle was displayed. The yellow square indicated that one or more variables were in the warning range, and the red triangle indicated that one or more variables were in the caution range.

2. **Elimination of multiple mouse clicks**. In the COBRA workstation, controllers must make multiple mouse clicks to perform each step in a pass plan. In the baseline system implemented in the test bed, each step is performed via a single mouse click.

3. **Voice Control**. The COBRA interface has no voice synthesis or recognition capability. In the baseline condition in this study, the controller was required to use the voice recognition system to select the pass plan to be executed and to select the data pages that were displayed.

<u>Enhanced</u>. The enhanced condition contained extensions to the baseline HSI. These additions are:

1. **Anomalous Data Presentation**. The ability to use the continuous monitor display to identify and present measurands that are outside their normal limits. The controller can activate this feature in one of three ways.

- Voice. The controller can say "Show me all the cautions" to have all of the measurands in the caution range displayed. Similarly, saying "Show me all the warnings" causes all of the measurands in the warning range to be displayed.
- Mouse. Clicking the left mouse button when the cursor is on the red triangle will cause all of the measurands in the caution range to be displayed. Clicking when the cursor is on the yellow square causes all of the measurands in the warning range to be displayed.
- Touch screen. Pressing on the red triangle or the yellow square has the same effect as does clicking the mouse button.

2. **Subsystem Diagrams**. The controller was able to display subsystem diagrams. These diagrams show the controller the state of all of the controls and the current values of the measurands. The controller can change the state of any control by pressing the corresponding button on the touch screen, clicking it with the left mouse button, or using an appropriate voice command.

3. **Data Displays**. The controller can have the value of a measurand displayed using a voice command. For example, if the controller was interested in the value of C+150V, then he could say "Show me the value of C+150V." The system would respond by displaying a dial gauge showing C+150V. This gauge also shows the nominal, warning, and caution ranges. This eliminates the need for the controller to scan data pages to obtain the current value of the measurand of interest. The graphic display of the value also shows trend information, which is difficult for humans to extract from a purely digital display.

4. **Voice To Operate Satellite Controls**. In the enhanced condition, the controller is allowed to utilize voice to operate the satellite controls. For example, if the controller wished to turn transmitter from 20 to 2.5 watts he could say something like "Set transmitter A to two and a half watts." This feature can be used in lieu of calling up and performing a pass plan.

5. **Synthesized Voice Feedback**. When a controller uses voice to control the satellite, the test bed responds by confirming the change using a synthesized voice. To continue the example above, after the controller has instructed the system to change the power setting the test bed would say "Set to two point five watts."

<u>SUBJECT</u>

A former satellite controller with over 7 years experience participated in this research. The controller was a man 37 years of age. The controller was paid for his participation. This controller was a former employee of MTI, and had been the principal investigator on this program during the Phase 1 effort.

METHOD

Scenarios. A total of eight scenarios were implemented in the test bed for this effort. These scenarios are labeled "Cobra 1" to "Cobra 8". The scenarios were developed jointly by CERES and AFRL previously, and were used here as a means to demonstrating the capabilities of the test bed

Procedure. Upon arrival at MTI's laboratory facility in Santa Clara California, the controller was first given a briefing. This briefing described the purposes of this effort, and informed of known risks. The controller was also advised that they had the right to terminate the experiment at any time.

Following the briefing, the controller was given a demonstration of the test bed. During this demonstration, the HSI features that would be used were pointed out. Questions from the controller were solicited and answered.

The briefing and demonstration required approximately one hour.

At this point, the experimenter's prepared the voice recognition system for this "new user". Once set up, the controller "trained" the voice recognition system. This training consisted of reading a variety of passages that were displayed on the screen. We have found that individual training of the voice system is often necessary in order to have an acceptable level of recognition accuracy. This training requires approximately one and a half hours to complete.

After the training of the voice recognition system had been completed, the experimenters configured the test bed for the first condition. This required copying the default states for that scenario into a file named *DefaultDataList.h*. This is done at the controller's workstation. On the experimenter's workstation, the experimenters set any anomalies required in that scenario.

Once the system was configured, the experimenter was given a script to follow. This script provided the controller information about the tasks that would be performed. This script contained information that a controller who is fully certified to perform a support on that satellite would know. The limits of measurands and the names of pass plans are examples. The script also contained relevant phrases that would be recognized by the voice system. While the provision of a script with this level of detail is not representative of the manner in which a controller would work in an operational environment, it allowed us to use an experienced controller without the need for extensive training on the particulars of this satellite. (This type of training often takes weeks of on-the-job training to accomplish.) Appendix 1 contains the scripts for all of the scenarios used here.

After becoming familiar with the script for that scenario, the controller was allowed to walk through that scenario. Each walk through was considered a training run and no data was collected. In a few instances, minor inconsistencies in terminology between the pass plan and data screens were noted. The experimenters provided guidance to the controller on how to deal with these on a case-by-case basis.

After the walk through was completed, any questions from the controller were answered. Then the controller repeated the scenario. During this run, data was collected.

The order of the runs was as follows:
Cobra 1 - Baseline
Cobra 1 - Enhanced
Cobra 2 - Baseline
Cobra 2 - Enhanced
Cobra 3 - Baseline
Cobra 3 - Enhanced
Cobra 4 - Baseline
Cobra 4 - Enhanced
Cobra 5 - Baseline
Cobra 5 - Enhanced
Cobra 6 - Baseline
Cobra 6 - Enhanced
Cobra 8 - Baseline
Cobra 8 - Enhanced

The Cobra 7 scenario was not run. During the first walk through an error in implementation was discovered and there was not sufficient time to correct the error during the time frame available.

Following completion of all of the test conditions, the controller completed a subjective questionnaire. This was followed by an open-ended discussion of the HSI and test bed. The questionnaire is shown is Appendix 2.

Results

As this was intended to be a demonstration of the test bed's capabilities and since only one subject participates, only descriptive statistics for a few important measures were computed. These computations were made to demonstrate the ability to import the raw data files into one commonly used spreadsheet. The results are presented below.

<u>TIME TO COMPLETE A SCENARIO.</u>

The amount of time that a controller requires to perform a scenario is one measure of how effective the HSI is. Operationally, time to complete a scenario is important for several reasons. First, in the case of one or more satellite anomalies, rapid action by the controller could result in the satellite being saved, or at least reduce the possibility of further degradation. Secondly, if controllers are able to perform supports more quickly, the amount of AFSCN resource time needed can be reduced. This would allow the system to support a greater number of satellites.

It is expected that a better HSI will allow the controller to complete a scenario more quickly than is possible with an inferior HSI, all else being equal. Here, it was expected that the "enhanced"

HIS would allow the controller to more quickly complete a scenario. This decrease in time to complete a scenario was expected occur because:

- The controller can use voice commands and controls on the system diagrams to directly change settings on the satellite, eliminating the need to go through a pass plan in a step-by-step fashion
- The controller can readily find the measurands that are outside their normal range without performing an item-by-item search.
- Feedback on the effects of actions taken is shown in the pass plan (if one is used), eliminating the need to scan a page of data to see if a state had changed.
- Feedback on voice commands that change the state of a variable is provided via synthesized voice, again eliminating the need for the controller to visually scan a page of data to see if the was accepted by the system. This feedback is not dependent on the controller's use of a pass plan.

Figure 1 shows the average time required for the controller to complete each scenario. Inspection of this figure shows that the time to complete a scenario averaged approximately 250 seconds in the baseline condition and just a bit over 100 seconds in enhanced condition. This result is in the predicted direction. The magnitude of the change is striking. However, it may be that these scenarios or the performance of this particular controller are not representative. Additional research would be needed to get a better estimate of the potential time savings that would be realized by using an improved HSI.

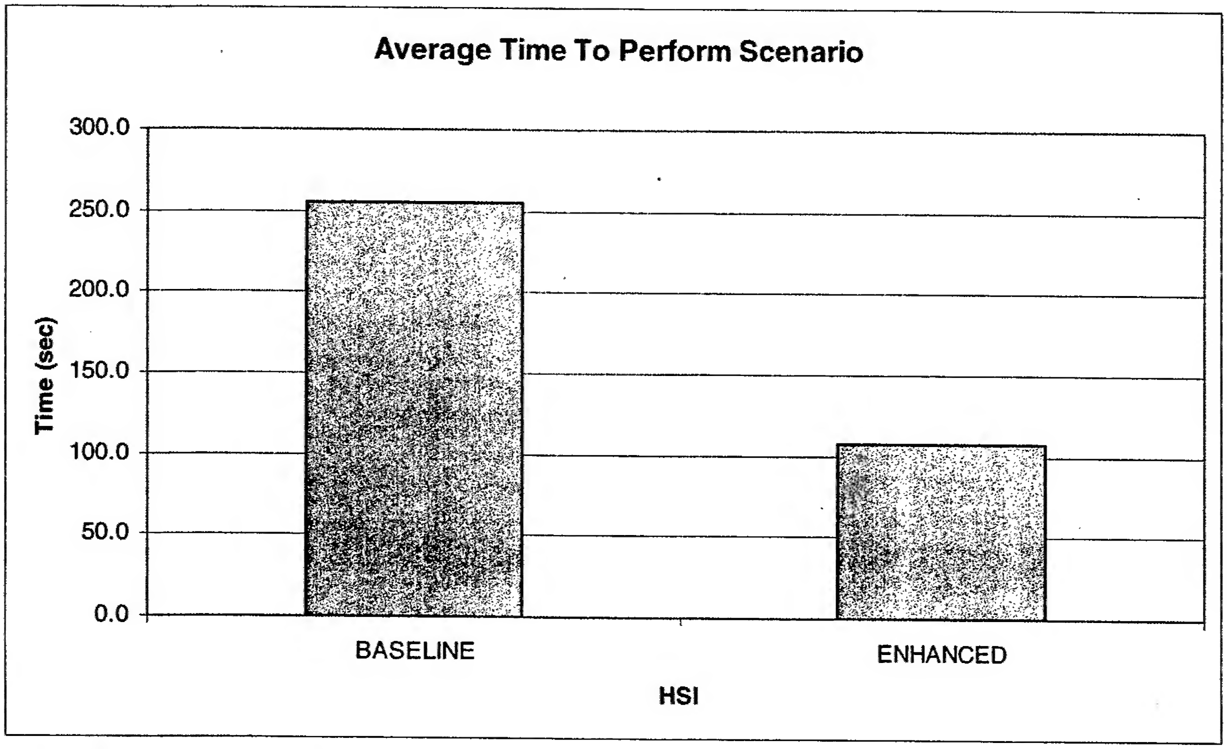


Figure 1. Average Time To Complete Scenarios.

AVERAGE NUMBER OF VOICE EVENTS

None of the currently operational satellite control systems that we are aware of have voice recognition capability. We expected that the number of voice events would be greater in the enhanced condition would be greater than in the baseline condition. Figure 2 shows the average number of voice events in the baseline and enhanced conditions.

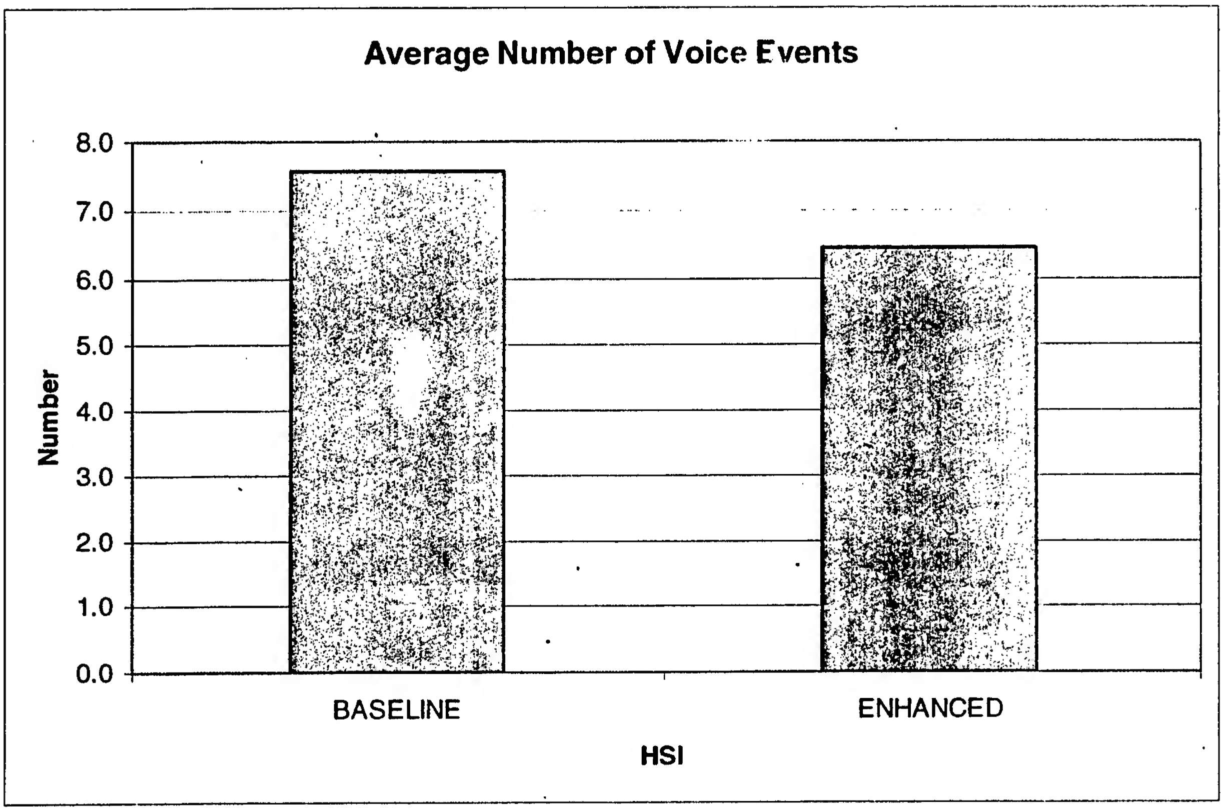


Figure 2. Average Number Of Voice Events.

Looking at this figure, it appears that there was about one more voice interaction in the baseline condition than in the enhanced condition. This result was unexpected. Closer examination of the data and the experimenter log sheets suggests that this result is due to the need of the controller to use the voice recognition system to access data pages. These pages are needed to verify the effects of the steps performed by the pass plan. If there had been no voice recognition system in the baseline condition, we expect that the direction of this difference would be reversed as the controllers would need to use a mouse or touch screen to select the pages for display. Similarly, the controllers used voice recognition system to call up a pass plan. If there were no voice recognition system in the baseline condition, then this would require a mouse click or touch screen press.

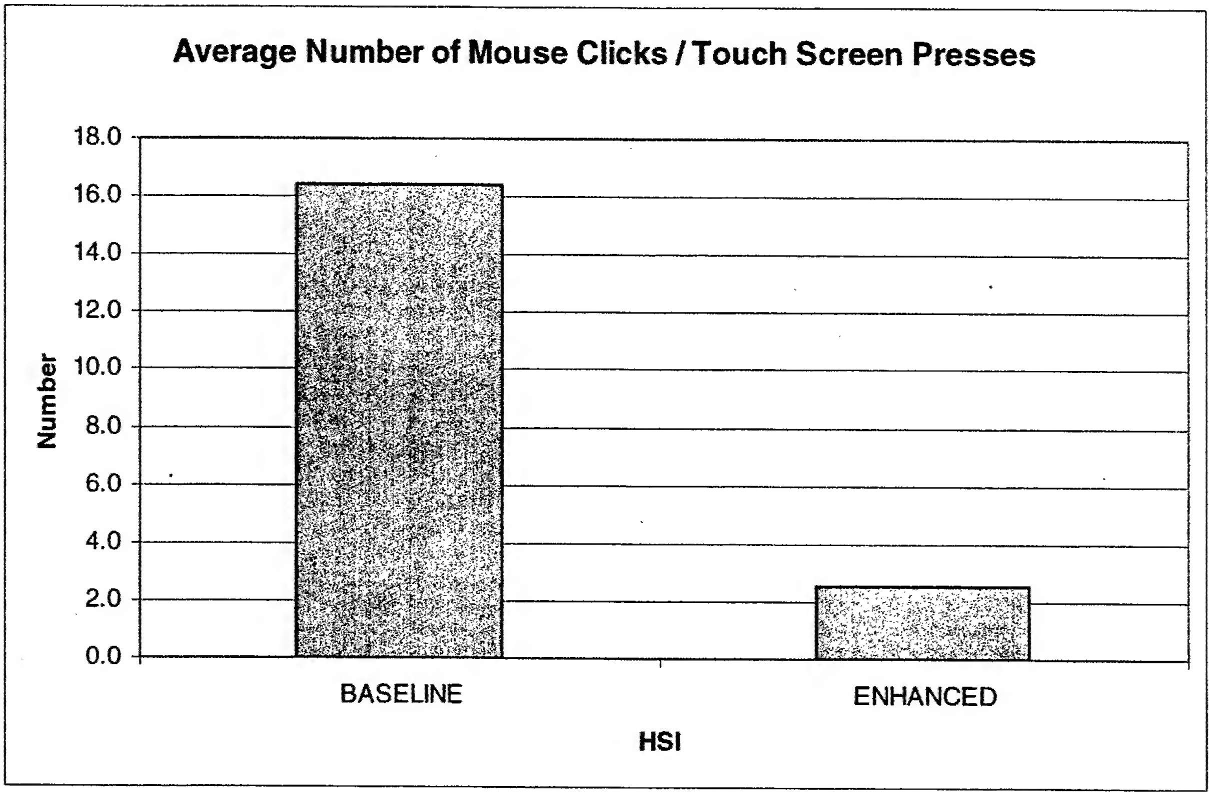
Therefore, it seems that this result is due to the need of the controllers to use the voice recognition system as a substitute for actions that would normally be done using a mouse in the baseline condition. We do not have data available here to determine whether or not the

controllers would use the mouse rather than voice to call up data pages and pass plans, due to the way in which the baseline system was implemented.

AVERAGE NUMBER OF MOUSE CLICKS AND TOUCHSCREEN PRESSES

It was expected that controllers would require a greater number of mouse clicks and/or touchscreen presses in the baseline condition than in the enhanced condition. This was expected because the only method to proceed through the pass plan is to "click" on each step. It is worth noting that in our implementation of the pass plans, we reduced the number of "clicks" required at each step from three in the COBRA workstation to one.

Figure 3 shows the average number of mouse clicks and touchscreen presses.



Examining this figure, it is clear that in the enhanced condition the controller had fewer mouse clicks than in the baseline condition. This result occurred even though the controller had the option to perform pass plans in the enhanced condition. This suggests to us that this controller had a preference for using voice commands when they were available, as opposed to selecting a process where a great deal of manual action was required. One implication of this may be that a touchscreen has only a small added value compared to a voice interface for user acceptance.

These data may artificially inflate the number of mouse clicks used in the enhanced condition. In the testbed, the voice recognition system only operates when the main window is in "focus".

That is, if another window, such as a system diagram, is highlighted then the voice system will not operate. The controller must click inside the main window whenever another window gains focus. (This is an issue in the Windows environment. It is very difficult, if not impossible, to keep the focus on one specific window while others are being opened. We have not developed a solution for this problem as yet.)

SUBJECTIVE RATINGS

After completing all of the scenarios, the controller was asked to provide ratings on a 7–point scale. On this scale a value of "1" indicated that the feature hindered the controller's performance, and a value of "7" indicated that the feature was extremely useful. Because only one controller participated in this exercise no analyses were performed. The ratings are presented here for completeness.

Question 1. This question was concerned with the continuous monitor displaying the presence of any measurands in the caution or warning regions. (The continuous monitor is the display in the upper portion of the left display. A yellow square indicated that one or more measurands was in the warning range, a red triangle indicated that one or more measurands were in the caution range. The presence of a green circle indicated that all measurands were in the nominal range.) The controller rated this feature a "7".

Question 2. Ability to have the system identify measurands in warning or caution states. (This was accomplished by pressing on the caution or warning indication, or stating "show me all the warnings" or "show me all the cautions" causes those measurands to be displayed.) The controller did not provide a rating for this, and indicated that he did not recall using this feature.

Question 3. Voice commands for calling up the desired pass plan. The controller rated this feature a "5."

Question 4. Voice commands for commanding the satellite (e.g., "turn the heater off"). The controller rated this feature a "6."

Question 5. Voice commands for requesting displays showing the values of measurands in warning or caution states (e.g., "Show me all the cautions"). The controller did not provide a rating for this feature and indicated that when he called up "all of the cautions" he did it using the touchscreen.

Question 6. Graphical display of the warning and caution ranges for measurands. (These ranges are displayed along with a pointer showing the current value of that measurand.) This feature was rated a "7" by the controller.

Question 7. Graphical display of the current value of measurands. A "5" rating was given by the controller for this feature.

Question 8. Presentation of the values or states to be verified in the pass plan (as opposed to presenting the values only in data pages). This feature was rated a "7."

Question 9. Touchscreens. The controller rated the touchscreens a "5."

Question 10. Overall, the use of speech commands, touch sensitive screens, and graphical display of information. The overall rating given by this controller was a "7."

After completing the questionnaire, the controller and the experimenters discussed the HSI features and the testbed. Below are summaries of points made by the controller during this discussion.

First, the controller liked the system diagrams most of all of the features. He felt that this presentation format offered a significant improvement over the existing data pages. (His term for the existing alphanumeric pages was "phonebook format.") The diagrams highlight the interdependence of the measures in a way that is meaningful to an experienced controller.

The controller went on to say that he felt uncomfortable with the ability to command a satellite by simply touching a button or making a voice command. He felt that some type of verification or check to insure that the command was the correct one would be necessary.

The voice and touch interfaces were very good. They would be particularly useful when attempting to identify and diagnose a problem with the satellite.

The voice command "Tell me the value of ...." would be very good. This command would respond by announcing the current value of that variable. This feature, the controller stated, "would be like having a personal assistant". We would like to point out that this command is implemented in the test bed. Unfortunately, because its operation is unreliable, we did not allow the controller to use this HSI feature in these scenarios. We are currently working with the vendor, Dragon Systems, to develop a solution to the reliability problem.

The automatic sensing and displaying of out-of-limits values was very valuable. He indicated that as a controller, a great deal of time was spent scanning data trying to detect values that are out-of-limits. Further, the ability to detect trends in a variable that indicates an emerging a problem, or patterns of variables that together indicate problems even though none are individually out of tolerance would be a nice tool to have.

The controller indicated that a way to get the recent history of a variable would be useful. This would help him identify trends that are leading to an out-of-tolerance situation. It would also help him diagnose a problem; "Did something fail suddenly, or did it drift off?" is an important question when diagnosing a problem. He indicated that in his experience controllers often relay on paper printouts in a binder to identify past values of variables. This is time consuming, and the patterns aren't always obvious from looking at the numerical data. A time history graph for one variable had bee implemented in the test bed previously. After the controller made this point, he was asked to bring up the graph. The controller indicated that this type of graph would be very useful to him.

Conclusion and Discussion

The principal goal of this effort was to demonstrate the suitability of the test bed in a human-system interface research environment. The ability to present the controller a set of scenarios based on satellite control tasks of interest to CERES and AFRL was demonstrated. During this effort the collection and analysis of performance data useful in assessing the effectiveness of two different HSIs was also demonstrated.

The HSIs used in this study both incorporate improvements over the COBRA interface. COBRA was selected as a model for this program because it is the most advanced HSI being used to support actual satellites of which we are aware. In this case, the baseline version is more similar to the COBRA HSI than is the enhanced HSI. These data do not allow direct comparison between the COBRA system and those tested here. However, it seems reasonable to expect that the performance with COBRA would be inferior to these systems tested here. (If for no other reason than three times as many mouse clicks would be required, and each mouse click requires a small amount of time. These times add up and would result in the support taking a greater amount of time.) The inference we draw from this is that improvements in the HSI used by satellite controllers would result in a decrease in the amount of time required to perform supports. This decrease would allow the satellite control network to reduce the time allotted for each support. This would free up these assets allowing either more supports of the same satellite, or support of additional satellites.

The difference in performance between the baseline and enhanced conditions in this study

As a new era in which space based assets are used in a more active role to support the warfighters, it is likely that increased supports will be required. During these supports, the satellite will be configured to best accomplish its tasking. In some cases, this may be retargeting sensors or the points on earth that are observed or otherwise serviced by that satellite. In other cases, satellites may take other actions to protect themselves or other assets. Increasing the ability of the system to accommodate the additional supports will be required by this new way of doing business. Improvements in the HSI can make this happen without major changes to the rest of the network.

Appendix 1 -Scenario Scripts

COBRA 1 – A SIDE POWER AMPLIFIER FAILED
BASELINE

Anomalies set by the experimenter
CPA1AW – Low Red
CPA1AV – Low Red

CONTROLLER TASKS

1. Find out if all of the measurands are in the normal range

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

The controller should identify CPA1AW and CPA1AV as being out of tolerance on the Communications System page.

On the Communications Status page note that EP1ASB is ON and EPP1BSB is OFF.

These failures indicate that the A-side power amp failed because of an electrical malfunction.

2. Determine the correct course of action.

In this case, the controller should turn OFF the A-side power amp and transmitter, and then turn ON the B-side power amp and transmitter.

3. Execute the appropriate pass plan

"Show me the pass plan for an A side power amp failure" or "Show me pass plan two"

At this point, the anomalies will be removed by the experimenter

4. Verify B-side transmitter and power amp are on. CPA1BW and CPA1BV should be in the nominal range.

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

COBRA 1 – A SIDE POWER AMPLIFIER FAILED
ENHANCED

Anomalies set by the experimenter
CPA1AW – Low Red
CPA1AV – Low Red

1. Determine that there is one or more anomalies. This is done by scanning the upper portion of the left display to see if there are any cautions or warnings.

"Show me the list of Cautions" or Touch the Red Triangle

2. Inspect the two gauges that appear.

3. Look at the link 1 subsystem diagram
"Show me link 1 communications flow"

4. Determine that these failures indicate that the A-side power amp failed because of an electrical malfunction.

5. Execute the appropriate pass plan
"Show me the pass plan for an A side power amp failure" or "Show me pass plan 2"

or

"Turn off EP1ASB"
"Turn Off ET1AOB"
"Set CSW3PB to B"
"Set CSW4PB to 20 watts"
"Set CSW5PB to 20 watts"
"Turn on ET1BOB"
"Turn on EP1BSB"

At this point, the anomalies will be removed by the experimenter

6. Verify all anomalies have been removed from the system by inspecting the status indicator bar on the left screen. Only green circles should be visible. The red triangle and yellow square should not be visible.

COBRA 2 – A SIDE TRANSMITTER FAILURE
BASELINE

Anomalies set by the experimenter
CCT1AW – Low Red
CCT1AV – Low Red

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range.

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

The controller should identify CCT1AW and CCT1AV as being out of tolerance on the Communications System page.

The controller should identify CCT1AW as being out of tolerance on the Communications Status Page.

The controller should identify CCT1AW and CCT1AV as being out of tolerance on the on the Bus page.

2. Select the appropriate pass plan

"Show me the pass plan for an A side two and a half watt transmitter failure" or **"Show me pass plan three"**

3. Verify that ET1AOB is turned off. This is on the communications status page.

4. Verify that CSW3PB is now set to the B channel. This is on the Communication System page or on the Communications Status page.

At this point the anomalies should be turned off by the experimenter.

5. Verify that ET1BOB is now on. This is on the Communications Status page.

COBRA 2 – A SIDE TRANSMITTER FAILURE ENHANCED

Anomalies set by the experimenter
CCT1AW – Low Red
CCT1AV – Low Red

1. Determine that there is one or more anomalies. This is done by scanning the upper portion of the left display to see if there are any cautions or warnings.

"Show me the list of Cautions" or Touch the Red Triangle

2. Inspect the two gauges that appear.

3. Look at the link 1 subsystem diagram

"Show me link 1 communications flow"

4. Determine that these failures indicate that the A-side transmitter has failed.

5. Execute the appropriate pass plan

"Show me the pass plan for an A side two and a half watt transmitter failure" or **"Show me pass plan three"**

or

"Turn off ET1AOB"
"Set CSW3PB to B"
"Set CSW4PB to two and a half watts"
"Set CSW5PB to two and a half watts"
"Turn on ET1BOB"

At this point, the anomalies will be removed by the experimenter

6. Verify all anomalies have been removed from the system by inspecting the status indicator bar on the left screen. Only green circles should be visible. The red triangle and yellow square should not be visible.

COBRA 3 – A SIDE TRANSMITTER FAILURE
BASELINE

Anomalies set by the experimenter
CCT1AW – Low Red
CCT1AV – Low Red

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range.

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

The controller should identify CCT1AW and CCT1AV as being out of tolerance on the Communications System page.

The controller should identify CCT1AW as being out of tolerance on the Communications Status Page.

The controller should identify CCT1AW and CCT1AV as being out of tolerance on the on the Bus page.

2. Execute the appropriate pass plan

"Show me the pass plan for an A side twenty watt transmitter failure" or **"Show me pass plan four"**

3. Verify that ET1AOB is turned off. This is on the communications status page.

4. Verify EP1ASB is off. This is on the communications status page.

5. Verify that CSW3PB is now set to the B channel. This is on the Communication System page or on the Communications Status page.

6. Verify CSW4PB is set to 20 watts. This is on the Communications System Configuration page or on the Communications Status Page

7. . Verify CSW5PB is set to 20 watts. This is on the Communications System Configuration page or on the Communications Status Page

8. Verify that ET1BOB is now on. This is on the Communications Status page.

At this point the anomalies should be turned off by the experimenter.

9. Verify EP1BSB is on. This is on the communications status page.

COBRA 3 – A SIDE TRANSMITTER FAILURE
ENHANCED

Anomalies set by the experimenter
CCT1AW – Low Red
CCT1AV – Low Red

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range. This is done by scanning the upper portion of the left display to see if there are any cautions or warnings.

"Show me the list of Cautions" or Touch the Red Triangle

2. Inspect the two gauges that appear.

3. Look at the link 1 subsystem diagram

"Show me link 1 communications flow"

4. Determine that these failures indicate that the A-side transmitter has failed.

5. Execute the appropriate pass plan

"Show me the pass plan for an A side twenty watt transmitter failure" or **"Show me pass plan four"**

or

Make the voice commands

"Turn off ET1AOB"
"Turn off EP1ASB"
"Set CSW3PB to B"
"Set CSW4PB to twenty watts"
"Set CSW5PB to twenty watts"
"Turn on ET1BOB"
"Turn on EP1BSB"

At this point the experimenter removes the anomalies.

6. Verify all anomalies have been removed from the system by inspecting the status indicator bar on the left screen. Only green circles should be visible. The red triangle and yellow square should not be visible.

COBRA 4 – B SIDE TRANSMITTER FAILURE
BASELINE

Anomalies set by the experimenter
CCT1BW – Low Red
CCT1BV – Low Red

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range.

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

The controller should identify CCT1BW as being out of tolerance on the Communications System page.

The controller should identify CCT1BW as being out of tolerance on the Communications Status Page.

The controller should identify CCT1BW and CCT1BV as being out of tolerance on the on the Bus page.

2. Execute the appropriate pass plan

"Show me the pass plan for a B side transmitter failure" or **"Show me pass plan five"**

3. Verify ET1BOB is off. This is on the Communications Status page.

4. Verify EP1BSB is off. This is on the Communications Status page.

5. Verify CSW3PB is set to A. This is on the Communication System Configuration page

6. Verify CSW4PB is set to 2.5 watts. This is on the Communication System Configuration page.

7. Verify CSW5PB is set to 2.5 watts. This is on the Communication System Configuration page.

The experimenter removes all anomalies at this point.

8. Verify ET1AOB is set to on. This is on the Communications Status Page.

COBRA 4 – B SIDE TRANSMITTER FAILURE
ENHANCED

Anomalies set by the experimenter
CCT1BW – Low Red
CCT1BV – Low Red

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range. This is done by scanning the upper portion of the left display to see if there are any cautions or warnings.

"Show me the list of Cautions" or Touch the Red Triangle

2. Inspect the two gauges that appear.

3. Look at the link 1 subsystem diagram

"Show me the link 1 communications flow"

4. Determine that these failures indicate that the B-side transmitter has failed.

5. Execute the appropriate pass plan
"Show me the pass plan for an A side twenty watt transmitter failure" or "Show me pass plan five"
or
Make the voice commands

"Turn off ET1BOB"
"Turn off EP1BSB"
"Set CSW3PB to A"
"Set CSW4PB to two and a half watts"
"Set CSW5PB to two and a half watts"
"Turn on ET1AOB"

At this point the experimenter removes the anomalies.

6. Verify all anomalies have been removed from the system by inspecting the status indicator bar on the left screen. Only green circles should be visible. The red triangle and yellow square should not be visible.

COBRA 5 – A SIDE DATA RATE UNIT FAILURE
BASELINE

Anomalies set by the experimenter
CEC5AV – Low Red

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range.

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

The controller should identify CECA5V as being out of tolerance on the Communications System page.

The controller should identify CECA5V as being out of tolerance on the Communications Status Page.

The controller should identify CECA5V as being out of tolerance on the on the Bus page.

2. Execute the appropriate pass plan

"Show me the pass plan for an A side data rate unit failure" or **"Show me pass plan six"**

3. Verify EDECAB is off and that EDECBB is on. These are on the Communications Status page and on the Bus page.

The experimenter removes the anomaly at this point.

4. Verify CECA5V and CECB5V are nominal. These are on the Communications Status page and on the Bus page.

COBRA 5 – A SIDE DATA RATE UNIT FAILURE
ENHANCED

Anomaly set by the experimenter
CEC5AV – Low Red

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range. This is done by scanning the upper portion of the left display to see if there are any cautions or warnings.

"Show me the list of Cautions" or Touch the Red Triangle

2. Inspect the gauge that appears.

3. Look at the link 1 subsystem diagram

"Show me link 1 communications flow"

4. Determine that these failures indicate that the A-side data rate unit has failed.

5. Execute the appropriate pass plan

"Show me the pass plan for an A side data rate unit failure" or **"Show me pass plan six"**

or

Make the voice commands
"Turn off EDECAB"
"Turn on EDECBB"

At this point the experimenter removes the anomaly.

6. Verify all anomalies have been removed from the system by inspecting the status indicator bar on the left screen. Only green circles should be visible. The red triangle and yellow square should not be visible.

COBRA 6 – CHANGE THE DATA RATE TO 1024
BASELINE

Anomalies set by the experimenter
None

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range.

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

2. Execute the appropriate pass plan

"Show me the pass plan to change the data rate from two oh four eight to one oh two four" or
"Show me pass plan seven"

3. Verify CDCAMB and CDCBMB are set to CLEAR. These are on the Communication System Configuration page and on the Communications System page.

COBRA 6 – CHANGE THE DATA RATE TO 1024 ENHANCED

Anomalies set by the experimenter
None

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range. This is done by scanning the upper portion of the left display to see if there are any cautions or warnings.

2. Select the appropriate pass plan.
"Show me the pass plan to change the data rate from two oh four eight to one oh two four" or
"Show me pass plan seven"

or

"set CDCAMB to bypass"
"set CDCBMD to bypass"

3. Verify CDCAMB and CDCBMB are set to CLEAR. These are on the Communication System Configuration page and on the Communications System page.

COBRA 7 – A SIDE CRYPTOGRAPH FAILURE
BASELINE

Anomalies set by the experimenter
None

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range.

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

2. Execute the pass plan to turn off the A-side cryptograph and turn on the B-side cryptograph.

"Show me the pass plan for an A side cryptograph failure" or **"Show me pass plan eight"**

3. Verify EKG1AB is off and EKG1BB is on. These are on the Communications System Configuration page and on the Bus page.

COBRA 7 – A SIDE CRYPTOGRAPH FAILURE
ENHANCED

Anomalies set by the experimenter
None

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range. This is verified by looking at the status indicators on the left screen. There should be a green circle, and no yellow square or red triangle. Alternatively, the controller could inspect the measurands.

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

2. Execute the pass plan to turn off the A side cryptograph and turn on the B side cryptograph.

"Show me the pass plan for an A side cryptograph failure" or **"Show me pass plan eight"**

or

"Turn off EKG1AB"
"Turn on EKG1BB"

3. Verify EKG1AB is off and EKG1BB is on. These are on the Communications System Configuration page and on the Bus page.

COBRA 8 – A SIDE POWER AMPLIFIER OVERHEATING
BASELINE

Anomalies set by the experimenter
CPA1AT – high red

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range.

"Show me the communications system configuration page"
"Show me the communications status page"
"Show me the bus page"

The controller should identify CPA1AT as being high. This is on the Communication System Configuration, Communications Status, and Bus pages.

The controller should also note that EP1AHB (the heater) is enabled. This is on the Communication System Configuration and Communications Status pages.

2. Execute the pass plan to turn off the heater for the power amplifier.

"Show me the pass plan for an A side heating problem" or **"Show me pass plan nine."**

3. Verify EP1AHB is disabled. This is on the Communication System Configuration and Communications Status pages.

The experimenter does NOT remove the anomaly. The temperature will remain above normal throughout this support and will decrease slowly. The next support would check the temperature.

COBRA 8 – A SIDE POWER AMPLIFIER OVERHEATING ENHANCED

Anomalies set by the experimenter
CPA1AT – high red

CONTROLLER TASKS

1. Find out if all the measurands are in the normal range. This is done by scanning the upper portion of the left display to see if there are any cautions or warnings, or by saying

"Show me the list of cautions" or by touching the red triangle

2. Select the appropriate pass plan.

"Show me the pass plan for an A side heating problem" or **"Show me pass plan nine"**

or

"Turn off EP1AHB"

3. Verify that the heater has been disabled. This is on the Communication System Configuration and Communications Status pages.

The experimenter does NOT remove the anomaly. The temperature will remain above normal throughout this support and will decrease slowly. The next support would check the temperature.

Appendix 2 – Post Experiment Subjective Rating Form

Interface Feature Ratings

You have had the opportunity to use a variety of Human-System Interface (HSI) features while performing simulated satellite control tasks. Please indicate the usefulness of the following HSI features by circling the best response

1. Continuous Monitor Displaying The Presence Of Any Measurands In The Caution Or Warning Regions. (This is the display in the upper portion of the left display. A yellow square indicates one or more measurands in the warning range, a red triangle indicates one or more measurands in the caution range. The presence of a green circle indicates all measurands are in the nominal range.)

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful

2. Ability Have System Identify Measurands In Warning Or Caution States. (Pressing on the caution or warning indication, or stating "Show me all the warnings" or "Show me all the cautions" causes those measurands to be displayed.)

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful

3. Voice Commands For Calling Up The Desired Pass Plan.

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful

4. Voice Commands For Commanding The Satellite. (e.g., "turn the heater off.")

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful

5. Voice Commands For Requesting Displays Showing The Values Of Measurands In Warning Or Caution States (e.g., "Show me all the cautions.")

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful

6. Graphical Display Of The Warning and Caution Ranges For Measurands. (These ranges are displayed along with a pointer showing the current value of that measurand.)

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful

7. Graphical Display Of The Current Value Of Measurands.

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful

8. Presentation Of The Values Or States To Be Verified In The Pass Plan (As Opposed To Presenting The Values Only In Data Pages).

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful

9. Touchscreens

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful

10. Overall, the use of speech commands, touch sensitive screens, and graphical display of information.

1 ------------- 2 ------------- 3 -------------- 4 ------------- 5 -------------- 6 ------------ 7

Hindered My Performance	No Effect	Extremely Useful